

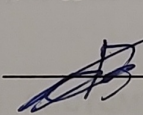
Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

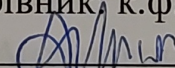
на тему: “Розробка системи захисту мовної інформації для проведення
конфіденційних розмов в автомобілі”

Виконав: студент 4 курсу, гр. 2КІТС-206
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

 Рижук В.В.

(прізвище та ініціали)

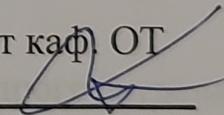
Керівник: к.ф.-м.н., доц., доцент каф. МБІС

 Шиян А.А.

(прізвище та ініціали)

« 6 » сервіс 2024 р.

Рецензент: к.т.н., доц., доцент каф. ОТ

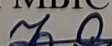
 Тарновський М.Г.

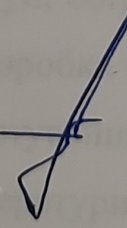
(прізвище та ініціали)

« 6 » сервіс 2024 р.

Допущено до захисту

Голова секції УБ кафедри МБІС

д.т.н., проф. Яремчук Ю.Є. 

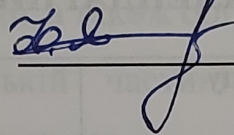
« 6 » сервіс 2024р. 

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти І-й (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ

Голова секції УБ, кафедра МБІС



д.т.н., проф. Яремчук Ю.Є.
“ 22 ” березня 2024 р.

З А В Д А Н Н Я

на бакалаврську дипломну роботу студенту

Рижук Влaдислaву Вікторoвичу

(прізвище, ім'я, по-батькові)

1. Тема роботи - Розробка системи захисту мовної інформації для проведення конфіденційних розмов в автомобілі

Керівник роботи: Шиян Анатолій Антонович, к.ф.-м.н., доцент, затверджені наказом вищого навчального закладу від “ 11 ” березня 2024 року № 80

2. Термін подання студентом роботи

3. Вихідні дані до роботи Windows; середовище програмування STM32CubeIDE; мови програмування для веб-розробки - C, Python.

4. Зміст текстової частини: вступ; обґрунтування вибору методу розробки та постановка задачі дослідження; розробка та схематичне зображення структури та алгоритмів роботи пристрою; тестування пристрою та програмного засобу; висновки; перелік використаної літератури; додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): порівняльний аналіз аналогів, структура прикладного програмного забезпечення, наукова новизна; метод та модель реалізації системи; обґрунтування доцільності розробки; тестування програми; публікація результатів роботи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I- III	Шиян А.А. к.ф.-м.н., доцент	22.03.2024	27.03.2024

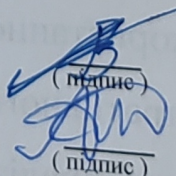
7. Дата видачі завдання 22 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Обґрунтування вибору методу розробки та постановки задач дослідження	27.03.2024	03.04.2024	
2	розробка структури та алгоритмів роботи програмного засобу	05.04.2024	15.04.2024	
3	Аналіз та вибір середовища програмування	16.04.2024	22.04.2024	
4	Розробка програмного засобу	23.04.2024	28.04.2024	
5	Тестування роботи розробленого програмного засобу	10.05.2024	17.05.2024	
6	Оформлення матеріалів до захисту БДР	18.05.2024	24.05.2024	

Студент

Керівник роботи

(підпис)

 (підпис)

Рижук В.В.
 (прізвище та ініціали)

Шиян А.А.
 (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 48 сторінок формату А4, на яких є 24 рисунки, 3 таблиць, список використаних джерел містить 30 найменувань.

Метою роботи є розробка системи захисту мовної інформації для конфіденційних розмов у автомобілі, що дозволить захистити конфіденційну інформації від несакціанованого доступу.

У загальній частині роботи розглянуто особливості побудови сучасних систем захисту мовної інформації, а також обґрунтована доцільність їх розробки. У розрахунково-конструкторській частині виконана розробка алгоритму програмного засобу. У технологічній частині розроблений програмний засіб автоматично виявляє початок конфіденційної розмови та виконую захист мовної інформації шляхом відтворення білого шуму. Проведено тестування програмного засобу, та зображено результати ефективності його роботи.

Ключові слова: захист інформації, мовна інформація, конфіденційна розмова, шумова завада.

ABSTRACT

The bachelor's thesis consists of 48 pages in A4 format, on which there are 24 figures, 3 tables, a list of words containing 30 names.

The method of work is the development of a system for protecting confidential information in the car, which allows you to protect confidential information from unauthorized access.

The final part of the work examines the peculiarities of current systems for the protection of mobile information, as well as the effectiveness of their development. The design and development part of Vikonan has developed a software algorithm. In the technological part of the software development, the software automatically detects the beginning of confidential information and protects private information by creating white noise. The software was tested and the results of the robot's effectiveness are shown.

Key words: information protection, language information, confidential conversation, noise interference.

ЗМІСТ

ВСТУП.....	7
1. ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	8
1.1. Аналіз стану питання розробки системи захисту мовної інформації для проведення конфіденційних розмов в автомобілі.	8
1.2. Аналіз існуючих методів.....	11
1.3 Аналіз існуючих реалізацій методів	15
1.4 Висновок	18
2. РОЗРОБКА ПРИСТРОЮ ЗАХИСТУ МОВНОЇ ІНФОРМАЦІЇ ДЛЯ ПРОВЕДЕННЯ КОНФІДЕНЦІЙНИХ РОЗМОВ У АВТОМОБІЛІ.....	20
2.1 Розробка структурної схеми	20
2.2 Вибір та обґрунтування елементної бази	23
2.3 Робота із базами даних	28
2.4 Висновок	30
3 РЕАЛІЗАЦІЯ ПРИСТРОЮ ЗАХИСТУ МОВНОЇ ІНФОРМАЦІЇ ДЛЯ ПРОВЕДЕННЯ КОНФІДЕНЦІЙНИХ РОЗМОВ У АВТОМОБІЛІ.....	31
3.1 Обґрунтування вибору мови програмування.....	31
3.2 Опис фрагментів лістингу програми.....	33
3.3 Тестування програмного засобу для автоматичного захисту мовної інформації	41
3.4. Висновок	46
ВИСНОВОК.....	47
ПЕРЕЛІК ПОСИЛАНЬ	48
ДОДАТКИ.....	51

ВСТУП

На сьогоднішній день безпека інформації стає все більш важливим аспектом у різних сферах діяльності, особливо коли мова йде про конфіденційність та захист персональних даних. З розвитком технологій та збільшенням кількості засобів для перехоплення та прослуховування, питання забезпечення безпеки мовної інформації набуває особливої актуальності. Однією з таких сфер, де необхідний надійний захист конфіденційних розмов, є автомобільний транспорт. Це стосується як приватних осіб, так і представників бізнесу та урядових структур, які часто використовують автомобілі для проведення важливих переговорів та зустрічей.

Мета роботи. Розробити систему захисту мовної інформації для проведення конфіденційних розмов в автомобілі.

Задачі роботи:

- Здійснити аналіз методів та засобів захисту мовною інформації при проведенні конфіденційних розмов в замкнутому середовищі;
- Розробити метод захисту мовної інформації для проведення конфіденційних розмов в автомобілі;
- Реалізувати засіб для системи захисту мовної інформації для *проведення конфіденційних розмов в автомобілі.

Об'єкт дослідження: Процеси захисту мовної інформації від прослуховування.

Предмет дослідження: Методи та засоби захисту мовною інформації при проведенні конфіденційних розмов в замкнутому середовищі.

Новизна роботи: Запропонована архітектура системи захисту мовної інформації, яка може бути реалізована в різних типах автомобілів.

Практична цінність роботи: Розроблена система захисту мовної інформації може бути використана для захисту конфіденційних розмов в автомобілі. [1]

1. ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1. Аналіз стану питання розробки системи захисту мовної інформації для проведення конфіденційних розмов в автомобілі.

Розробка систем захисту мовної інформації для автомобілів є актуальним напрямком досліджень і впроваджень у контексті сучасних вимог безпеки. Розглянемо основні аспекти, стан технологій та існуючі рішення в цій сфері.

1) Технічні засоби захисту



Рисунок 1 - Технічні засоби захисту

1.1. Шумоізоляція та звукові бар'єри:

Автомобілі високого класу вже використовують покращені матеріали для шумоізоляції, які можуть зменшити можливість перехоплення розмов ззовні. Спеціальні акустичні матеріали поглинають звук і знижують рівень шуму всередині кабіни. [2]

1.2. Активні системи шумозаглушення:

Технології активного шумозаглушення (Active Noise Cancellation, ANC) можуть використовуватися для приглушення фонових шумів і перешкоджання перехопленню розмов. Такі системи працюють шляхом генерування звукових хвиль, які гасять небажаний шум.

1.3. Вібраційні панелі:

Спеціальні панелі, які створюють вібрації на поверхнях автомобіля, можуть унеможливити використання лазерних мікрофонів, які вловлюють звукові вібрації зі скла автомобіля для прослуховування розмов.

2) Програмні та криптографічні рішення



Рисунок 2 - Програмні та криптографічні рішення

2.1. Шифрування звуку:

Існують рішення, які дозволяють шифрувати голосові дані, передані через автомобільні системи зв'язку. Це може бути корисно під час використання телефонних дзвінків або інших засобів комунікації в автомобілі. Наприклад, використання криптографічних протоколів типу SRTP (Secure Real-time Transport Protocol) для захисту VoIP-зв'язку.

2.2. Автентифікація та контроль доступу:

Системи контролю доступу та автентифікації користувачів можуть забезпечити захист від несанкціонованого використання автомобільних систем зв'язку. Це може включати біометричну автентифікацію (відбитки пальців, розпізнавання обличчя) для доступу до конфіденційних розмов.

3) Організаційні заходи



Рисунок 3 - Організаційні заходи

3.1. Обмеження доступу до автомобіля:

Фізичний контроль доступу до автомобіля і його систем, включаючи обмеження на використання мобільних пристроїв під час конфіденційних розмов, може знизити ризик несанкціонованого прослуховування. [3]

3.2. Навчання користувачів:

Проведення навчання для водіїв та пасажирів щодо безпечного використання автомобільних систем і виявлення потенційних загроз може значно підвищити рівень безпеки.

4) Існуючі рішення та розробки

4.1. Cooon Secure:

Ця система включає в себе різні технології для забезпечення конфіденційності розмов в автомобілі, такі як шумоізоляція, активне шумозаглушення та шифрування звуку.

4.2. Clear Voice™:

Технологія, що використовує передові методи шумозаглушення та шифрування звуку для забезпечення конфіденційності розмов у автомобілях. Ця система також забезпечує високу якість зв'язку при мінімальному фоновому шумі.

5) Виклики та перспективи

5.1. Технічні виклики:

Основні технічні виклики включають забезпечення високої якості захисту без значного погіршення користувацького досвіду. Наприклад, системи шумозаглушення повинні ефективно приглушувати небажані шуми без впливу на чіткість розмови.

5.2. Інтеграція з існуючими системами:

Інтеграція нових систем захисту з існуючими автомобільними системами може бути складним завданням, що вимагає ретельного проектування та тестування.

5.3. Економічна доцільність:

Висока вартість впровадження сучасних технологій захисту може бути перешкодою для їх масового використання. Тому важливо знаходити баланс між рівнем захисту та економічною доцільністю впровадження.

Аналіз стану питання показує, що існує значний прогрес у розробці та впровадженні систем захисту мовної інформації для автомобілів. Однак, виклики технічного, економічного та організаційного характеру вимагають подальших досліджень та інновацій. Сучасні технології вже пропонують ефективні рішення, але необхідно постійно вдосконалювати їх для забезпечення максимальної безпеки конфіденційних розмов в умовах зростання кіберзагроз та технологічного прогресу. [4]

1.2. Аналіз існуючих методів

Для вибору оптимальної системи захисту мовної інформації важливо провести порівняльний аналіз існуючих рішень. Розглянемо кілька відомих систем і їхні характеристики для визначення їх переваг і недоліків.

Основні критерії порівняння: рівень захисту, технології захисту, інтеграція з існуючими системами, вартість, зручність використання.

Аналіз систем

1. Cocoon Secure

Рівень захисту: високий рівень захисту завдяки використанню комбінованих технологій.

Технології захисту: шумоізоляція, активне шумозаглушення (ANC), шифрування звуку, вібраційні панелі.

Інтеграція з існуючими системами: система інтегрується з більшістю сучасних автомобілів, але може вимагати спеціалізованого встановлення.

Вартість: висока, оскільки включає комплексні рішення та передові технології.

Зручність використання: зручна в експлуатації, але може вимагати налаштування професіоналами.

Переваги: комплексний підхід до захисту, висока ефективність.

Недоліки: висока вартість, складність установки. [21]

2. Clear Voice

Рівень захисту: високий, завдяки використанню передових методів шумозаглушення і шифрування.

Технології захисту: активне шумозаглушення (ANC), шифрування звуку.

Інтеграція з існуючими системами: легка інтеграція з сучасними автомобільними системами зв'язку.

Вартість: середня, що робить її доступнішою для більш широкого кола користувачів.

Зручність використання: високий рівень зручності, оскільки система автоматизована і не вимагає складних налаштувань.

Переваги: доступність, легка інтеграція та використання.

Недоліки: менший спектр захисних технологій порівняно з Cocoon Secure.

3. Acousticom

Рівень захисту: середній, зосереджений на шумоізоляції та основних методах шумозаглушення.

Технології захисту: шумоізоляція, пасивне шумозаглушення.

Інтеграція з існуючими системами: вимагає мінімальних змін в існуючих автомобільних системах.

Вартість: низька, що робить її доступною для масового ринку.

Зручність використання: дуже зручна, не вимагає спеціальних налаштувань або обслуговування.

Переваги: низька вартість, простота встановлення та використання.

Недоліки: обмежений рівень захисту, менш ефективна в умовах високого рівня загроз. [24]

4. SecureComm

Рівень захисту: високий, завдяки використанню передових криптографічних технологій.

Технології захисту: шифрування звуку, криптографічні протоколи для VoIP.

Інтеграція з існуючими системами: добре інтегрується з системами, що підтримують VoIP та інші цифрові комунікації.

Вартість: від середньої до високої залежно від обраних функцій.

Зручність використання: зручна для технічно підготовлених користувачів, але може вимагати спеціалізованої підтримки.

Переваги: високий рівень захисту за рахунок шифрування, гнучкість у налаштуваннях.

Недоліки: може бути складною для не підготовлених користувачів, потребує підтримки з боку ІТ-фахівців. [25]

Таблиця 1. Порівняльна

Система	Рівень захисту	Технології захисту	Інтеграція	Вартість	Зручність використання	Переваги	Недоліки

Cocoon Secure	Високий	Шумоізоляція, ANC, шифрування, вібраційні панелі	Складна	Висока	Висока	Комплексний підхід	Висока вартість, складність установок
Clear Voice	Високий	ANC, шифрування	Легка	Середня	Висока	Доступність, легка інтеграція	Обмежений спектр технологій
Acousticom	Середній	Шумоізоляція, пасивне шумозаглушення	Мінімальна	Низька	Дуже висока	Низька вартість, простота	Обмежений рівень захисту
Secure Comm	Високий	Шифрування, криптографічні протоколи	Добра	Середня/висока	Середня/висока	Високий рівень захисту, гнучкість	Складна для не підготовлених користувачів

На основі проведеного аналізу, вибір оптимальної системи залежить від конкретних потреб та умов використання.

Cocoon Secure підходить для ситуацій, де потрібен максимально високий рівень захисту і є готовність інвестувати у складне і дороге рішення.

Clear Voice забезпечує високий рівень захисту за помірну вартість і підходить для користувачів, які потребують легкої інтеграції і зручності використання.

Acousticom є хорошим варіантом для масового ринку з обмеженим бюджетом, забезпечуючи базовий рівень захисту.

SecureComm ідеально підходить для технічно підготовлених користувачів і організацій, що вимагають високого рівня криптографічного захисту.

Вибір конкретної системи залежить від специфіки завдань, бюджету та технічних можливостей користувачів. [6]

1.3 Аналіз існуючих реалізацій методів

Аналіз існуючих реалізацій методів захисту мовної інформації в автомобілі включає як програмні рішення, так і апаратні пристрої. Ось огляд деяких з них з кількісними характеристиками, потрібною інформацією та недоліками

Основні методи розв'язання задачі

Апаратні пристрої

Шумоподавлюючі пристрої (Noise Generators)

Приклад: Speech Privacy Systems - Sound Masking Devices

Кількісні характеристики:

Рівень шуму: 45-60 дБ

Частота звуку: 500-4000 Гц

Розміри: різні моделі, середні розміри близько 15x15x5 см

Потрібна інформація: Розташування в автомобілі, джерело живлення, можливість регулювання рівня шуму.

Недоліки: Може створювати дискомфорт для водія та пасажирів, не забезпечує захист від зйомок. [26]

Акустичні сейфи (Speech Privacy Systems)

Приклад: Hemi Anechoic Chambers

Кількісні характеристики:

Рівень ізоляції: до 60 дБ

Матеріали: звукопоглинаючі панелі

Вартість: висока, від \$5000

Потрібна інформація: Розміщення в автомобілі, інтеграція з іншими системами.

Недоліки: Дорого, займає багато місця, потребує спеціальної установки.[5]

Мобільні глушники (Mobile Jammers)

Приклад: Portable Cell Phone Jammers

Кількісні характеристики:

Діапазон: до 30 метрів

Вартість: \$100-\$500

Живлення: акумулятор або автомобільний адаптер

Потрібна інформація: Діапазон частот, які блокує, тривалість роботи від акумулятора.

Недоліки: Може бути незаконним у деяких країнах, блокує всі мобільні пристрої у радіусі дії, що може бути незручно. [27]

Програмні продукти

Захищені мобільні телефони (Secure Mobile Phones)

Приклад: Silent Circle (Blackphone)

Потрібна інформація:

Платформа: Android

Шифрування: end-to-end, AES-256

Вартість: близько \$700 за пристрій

Недоліки: Висока вартість, обмежена функціональність порівняно зі звичайними смартфонами.[28]

Мобільні додатки для шифрування голосу

Приклад: Signal, Telegram

Потрібна інформація:

Платформа: iOS, Android

Шифрування: end-to-end, AES-256, Perfect Forward Secrecy (PFS)

Вартість: безкоштовно

Недоліки: Потребують інтернет-з'єднання, можуть бути обмежені державними регуляціями. [29]

Звукові кабінки (Soundproof Cabins)

Приклад: Mobile Soundproof Booths

Кількісні характеристики:

Рівень ізоляції: до 70 дБ

Матеріали: звукоізолюючі панелі, двері

Вартість: \$1000-\$5000

Потрібна інформація: Розміри, спосіб встановлення, джерело живлення.

Недоліки: Дорого, обмежений простір для використання в автомобілі.[30]

Таблиця 2. Порівняльний аналіз

Метод	Кількісні характеристики	Потрібна інформація	Недоліки
Noise Generators	45-60 дБ, 500-4000 Гц	Розташування, джерело живлення	Дискомфорт, не захищає від зйомок
Speech Privacy Systems	Ізоляція до 60 дБ, \$5000	Розміщення, інтеграція	Дорого, займає місце
Mobile Jammers	Діапазон до 30 м, \$100-\$500	Діапазон частот, тривалість роботи	Незаконно в деяких країнах, блокує всі пристрої
Secure Mobile Phones	Android, AES-256, \$700	Платформа, метод шифрування	Висока вартість, обмежена функціональність
Encryption Voice Apps	iOS, Android, AES-256, безкоштовно	Платформа, метод шифрування	Потребують інтернету, можуть бути обмежені
Soundproof Cabins	Ізоляція до 70 дБ, \$1000-\$5000	Розміри, спосіб встановлення	Дорого, обмежений простір

Noise Generators	45-60 дБ, 500-4000 Гц	Розташування, джерело живлення	Дискомфорт, не захищає від зйомок
------------------	--------------------------	--------------------------------------	---

Кожен метод захисту мовної інформації має свої переваги та недоліки, і їх ефективність може варіюватися залежно від конкретних умов та потреб користувача. Оптимальним підходом є комплексне використання кількох методів одночасно, що дозволить забезпечити найвищий рівень захисту конфіденційних розмов в автомобілі.

1.4 Висновок

Шумоподавлюючі пристрої та мобільні глушники ефективні для швидкого та відносно недорогого рішення, але можуть мати юридичні та практичні обмеження.

Акустичні сейфи та звукові кабінки забезпечують високий рівень захисту, але дорогі і важкі для впровадження в обмеженому просторі автомобіля.

Захищені мобільні телефони та додатки для шифрування пропонують зручне та доступне рішення для більшості користувачів, але залежать від наявності інтернету і можуть бути піддані обмеженням в деяких країнах.

Постановка задачі: Розробити ефективну систему захисту мовної інформації для проведення конфіденційних розмов в автомобілі, яка буде поєднувати апаратні та програмні методи захисту, враховуючи фінансові та просторові обмеження, а також відповідність законодавству.

Для цього необхідно виконати аналіз вимог користувачів:

З'ясувати основні вимоги до захисту мовної інформації для конкретних користувачів (наприклад, бізнесменів, політиків, юристів).

Визначити рівень конфіденційності та частоту проведення конфіденційних розмов.

Оцінити умови експлуатації:

Дослідити специфічні умови експлуатації системи в автомобілі (розміри, акустичні характеристики, джерела живлення).

Врахувати можливість інтеграції з існуючими системами безпеки автомобіля.

Обрати методи захисту:

Визначити найбільш підходящі апаратні та програмні методи захисту на основі проведеного аналізу.

Оцінити можливість комбінації методів для досягнення максимального рівня захисту.

Проектування системи:

Розробити архітектуру системи, яка включатиме обрані методи захисту.

Визначити необхідні апаратні компоненти та програмне забезпечення.

Розробка та тестування:

Реалізувати прототип системи та провести тестування розробки.

Оцінити ефективність системи та внести необхідні корективи.

2. РОЗРОБКА ПРИСТРОЮ ЗАХИСТУ МОВНОЇ ІНФОРМАЦІЇ ДЛЯ ПРОВЕДЕННЯ КОНФІДЕНЦІЙНИХ РОЗМОВ У АВТОМОБІЛІ

2.1 Розробка структурної схеми

Першим кроком потрібно визначити функціонал приладу, який буде реалізуватися. Основною функцією даного приладу є автоматичне виявлення та маскування конфіденційної розмови. Оскільки мовна інформація під час конфіденційної розмови може бути підслухана та використана у зловмисних цілях. Захист такої інформації відбувається різними способами і засобами. Одним з таких способів є пристрій маскування мовної інформації.

Даний прилад має багато переваг над простими системами захисту інформації. Система поєднує в собі аналіз аудіо і шифрування інформації використовуючи розроблений пристрій і акустичну систему машини.

Перевагою приладу є ціна і простота використання. При низьких затратах для створення даної системи, вона покриває велику кількість задач поставлених на неї. Дана система легко інтегрується до вже існуючих систем захисту мовної інформації. Також в даній системі використовується контролер з потенціалом для додавання функціоналу і подальшої модернізації.

Перший етап у створенні приладу полягає у розробці структурної моделі, яка визначає, з яких компонентів буде складатися пристрій і які зв'язки будуть між ними. Це дозволяє отримати загальне уявлення про складові частини приладу та їх взаємодію.

Наступним етапом є вибір елементної бази. Це включає визначення основного контролера, мікрофону, генератору шуму та інших необхідних елементів. Після цього необхідно створити принципову схему приладу. Принципова схема відображає зв'язки між різними радіокомпонентами і показує, як різні модулі взаємодіють між собою.

Ця схема надає уявлення про прилад в цілому, демонструючи його функціональні можливості захисту мовної інформації, основні блоки, вузли, частини та ключові зв'язки між ними. Після завершення цього етапу стає

зрозуміло, для чого потрібний даний пристрій і як він працює, а також як взаємодіють його складові частини.

Другим кроком потрібно зробити структурну схему приладу для забезпечення конфіденційності розмов у автомобілі. На схемі показано основні блоки приладу та зв'язки між ними: контролер, мікрофон, шумогенератор, аудіоперемикач, підсилювач, стабілізатор напруги, інтерфейс до автомобільної аудіосистеми та інше.

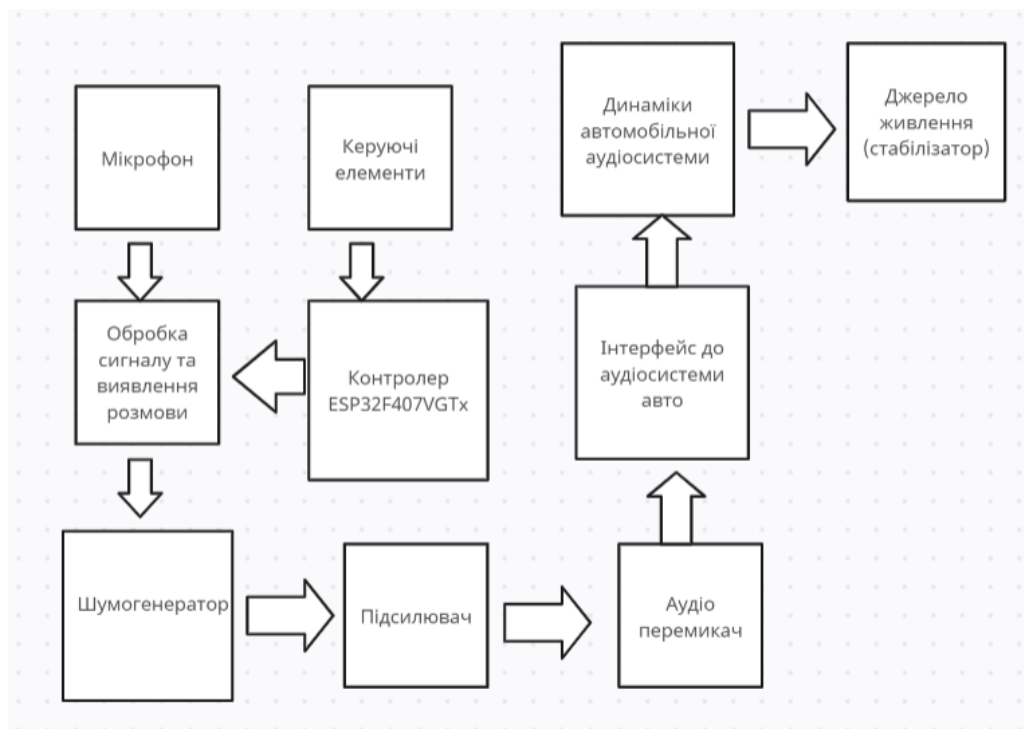


Рисунок 4 – Структурна схема приладу

Опис блок-схеми роботи системи(рисунок 5)

Початок роботи: Початок роботи пристрою, включення живлення.

Запис аудіо сигналу з мікрофона: Мікрофон записує звуки всередині автомобіля.

Аналіз аудіо сигналу на виявлення конфіденційної розмови: Контролер аналізує записаний аудіо сигнал на наявність конфіденційної розмови.

Виявлення конфіденційної розмови:

Так: Якщо конфіденційна розмова виявлена, переходить до наступного кроку.

Ні: Якщо конфіденційна розмова не виявлена, повертається до запису аудіо сигналу.

Генерація маскуючого шуму: Генерація маскуючого шуму для перекриття конфіденційної розмови.

Відтворення маскуючого шуму через аудіосистему автомобіля: Відтворення згенерованого шуму через динаміки автомобіля.

Перемикання аудіо сигналу на штатний, якщо конфіденційна розмова завершена: Перемикання аудіо сигналу на штатний сигнал автомобіля після завершення конфіденційної розмови.

Продовження запису та аналізу аудіо сигналу: Повернення до запису та аналізу аудіо сигналу для виявлення нових конфіденційних розмов.

Кінець роботи: Завершення роботи пристрою, вимкнення живлення або перехід у режим очікування.

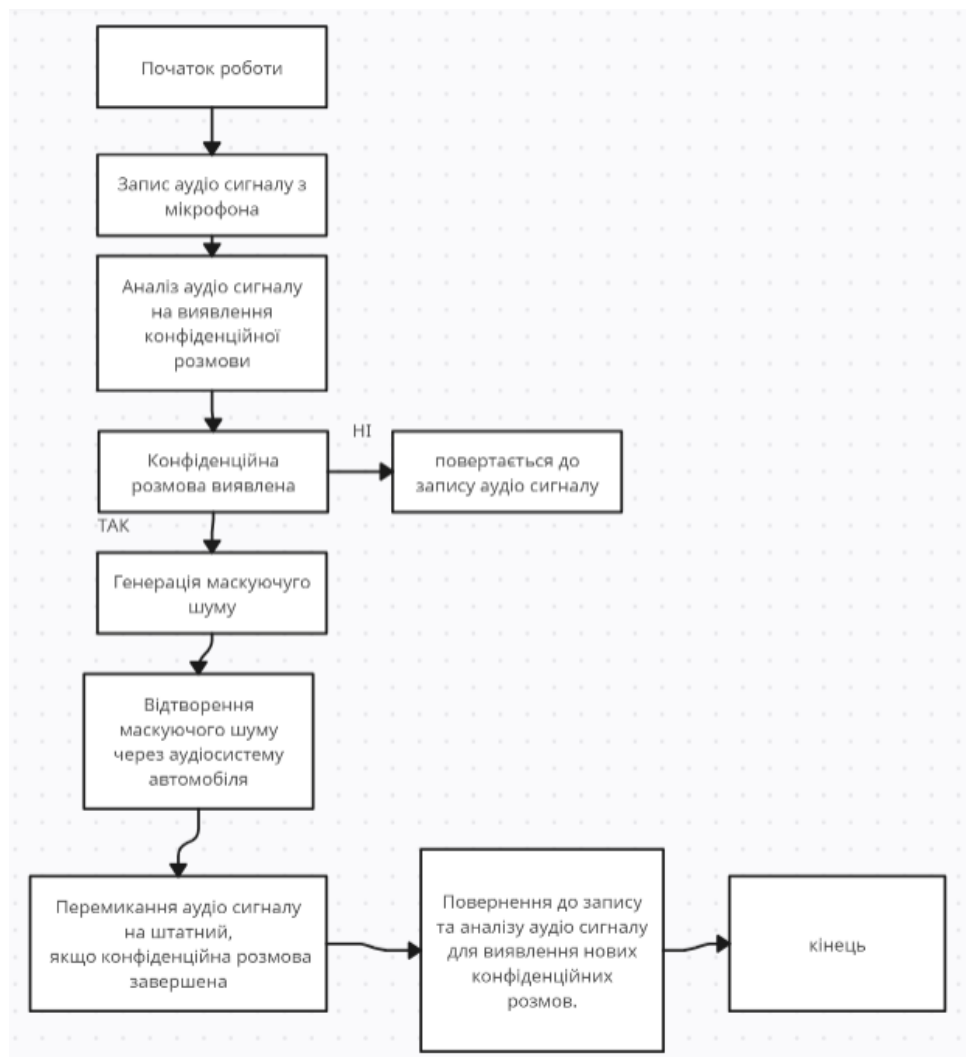


Рисунок 5 – Блок схема роботи приладу

2.2 Вибір та обґрунтування елементної бази

Мікроконтролер (STM32F407VGTx)

Причини вибору:

Потужність і продуктивність: STM32F407VGTx має високопродуктивний процесор ARM Cortex-M4 з частотою до 168 МГц, що дозволяє виконувати складні обчислення та обробку сигналів у реальному часі.

Великий обсяг пам'яті: 1 МБ флеш-пам'яті та 192 КБ оперативної пам'яті дозволяють зберігати великі програми та дані для аналізу аудіосигналів.

Розширені периферійні можливості: Мікроконтролер має багато вбудованих периферійних модулів, включаючи ADC, UART, SPI, I2C, що дозволяє легко підключати різні компоненти, такі як мікрофон, SD-карта, та інші периферійні пристрої.



Рисунок 6 - Зовнішній вигляд STM32F407VGTx

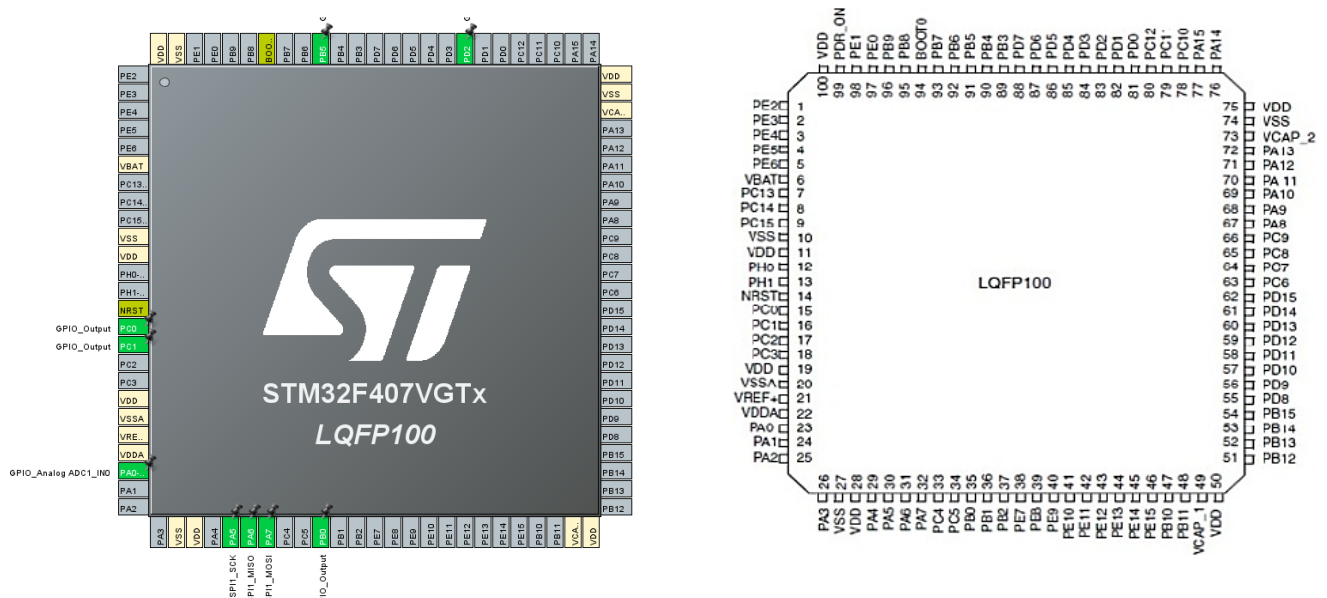


Рисунок 7 - Зображення виходів мікроконтролера STM32F407VGTx

Вбудована підтримка DSP: Підтримка цифрової обробки сигналів (DSP) допомагає у розпізнаванні ключових слів та обробці аудіосигналів.

Мікрофон (MAX9814)

Причини вибору:

Автоматичне регулювання підсилення (AGC): MAX9814 оснащений функцією автоматичного регулювання підсилення, що дозволяє отримувати стабільний аудіосигнал незалежно від рівня гучності джерела звуку.

Висока чутливість і якість звуку: Мікрофон має низький рівень шуму і високу чутливість, що забезпечує якісне захоплення звуку для подальшого аналізу.

Простота підключення: Мікрофон легко інтегрується з мікроконтролером через аналоговий вихід, що зручно для обробки сигналу за допомогою ADC.

Прошивається STM32F407VGTx за допомогою USB-to-Serial адаптеру на базі чіпа FT232RL. Для підключення використовуються виходи:

Піни для підключення:

TX адаптера -> RX мікроконтролера (PA10)

RX адаптера -> TX мікроконтролера (PA9)

GND адаптера -> GND мікроконтролера

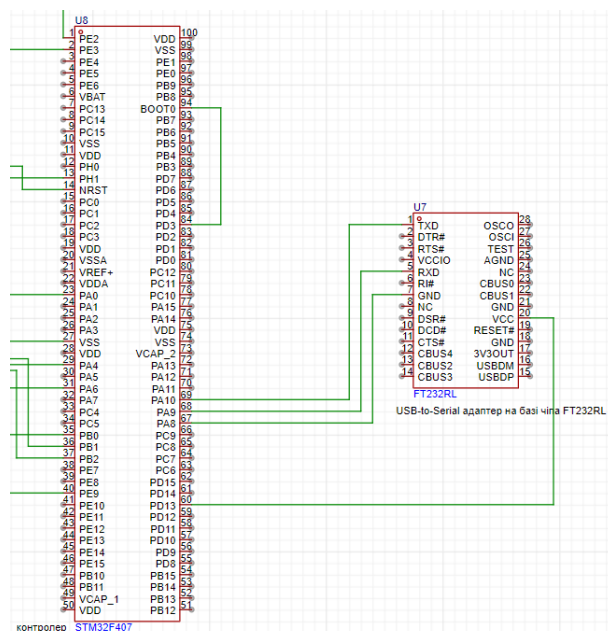


Рисунок 8 - Схематичне зображення підключення STM32F407VGTx до USB-to-Serial адаптеру

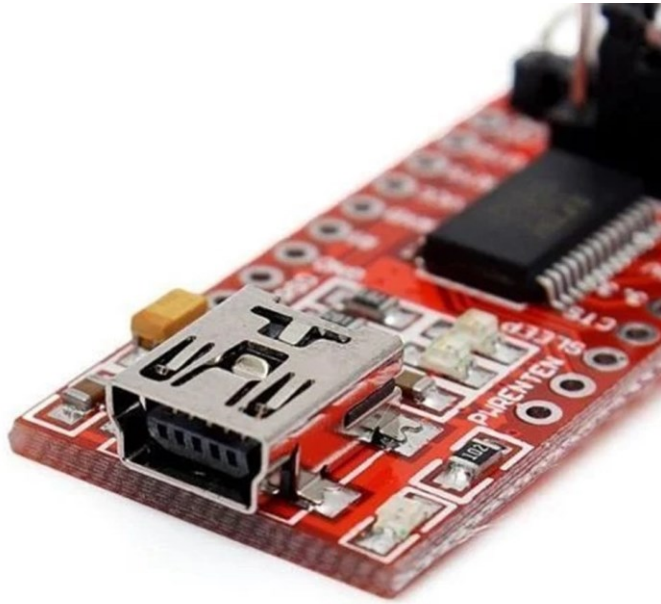


Рисунок 9 - USB-to-Serial адаптер на базі чіпа FT232RL

Шумогенератор (LTC6900CS5#TRMPBF, 20MHz)

Причини вибору:

Простота налаштування частоти: LTC6900 дозволяє легко налаштувати частоту генерації шуму шляхом підбору відповідних резисторів.

Стабільність і надійність: Генератор забезпечує стабільний вихідний сигнал з мінімальними відхиленнями, що важливо для створення ефективного білого шуму.

Малий розмір і низька споживана потужність: Цей компонент компактний і має низьке енергоспоживання, що підходить для використання в автомобільних системах.

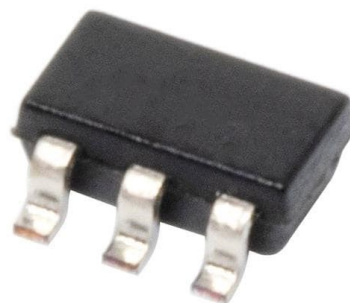


Рисунок 10 - Шумогенератор LTC6900CS5#TRMPBF

Аудіо перемикач (74НС4052)

Причини вибору:

Множинні канали перемикання: 74НС4052 забезпечує можливість комутації між декількома аудіосигналами, що дозволяє легко перемикати вихідні канали для відтворення шуму або звичайного аудіо.

Широка сумісність: Перемикач підтримує роботу з різними рівнями напруги, що забезпечує сумісність з іншими компонентами системи.

Низький опір і швидка комутація: Низький опір в замкнутому стані і швидка комутація дозволяють зберегти якість аудіосигналу без затримок.

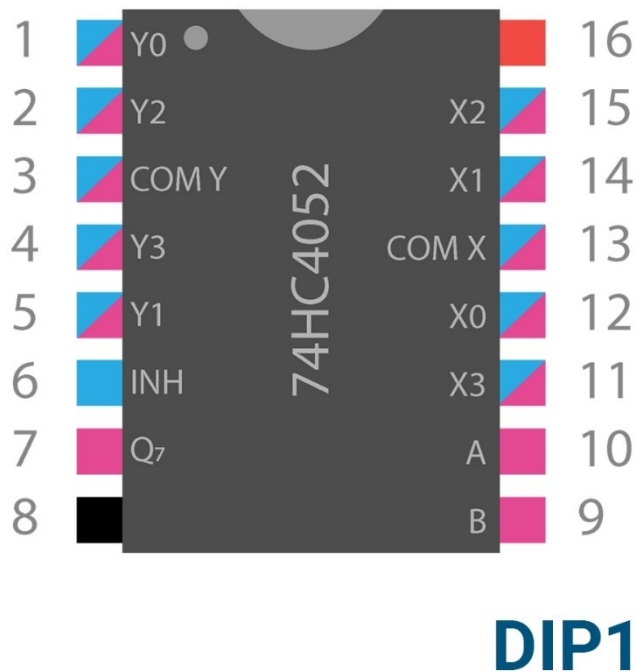


Рисунок 11 - Аудіо перемикач 74НС4052

Підсилювач (ТРА3116D2)

Причини вибору:

Висока потужність: Підсилювач забезпечує вихідну потужність до 50 Вт на канал, що достатньо для подачі білого шуму через автомобільні динаміки з достатньою гучністю.

Висока ефективність: Підсилювач має високу ефективність до 90%, що зменшує нагрівання і споживання енергії, що важливо для автомобільних систем.

Низький рівень спотворень: Забезпечує чисте та якісне звучання без значних спотворень, що важливо для ефективного заглушення конфіденційних розмов.

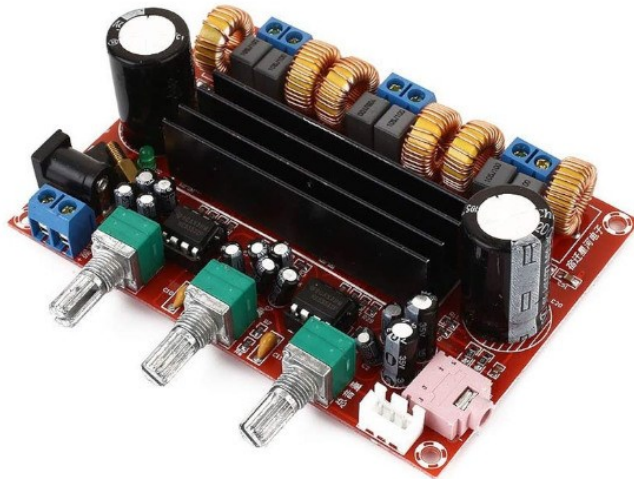


Рисунок 12 - Підсилювач (TPA3116D2)

Стабілізатор напруги (LM2596S-3.3 P+)

Причини вибору:

Стабільне вихідне напруження: LM2596S-3.3 P+ забезпечує стабільне вихідне напруження 3.3 В, що необхідно для живлення мікроконтролера та інших компонентів.



Рисунок 13 - Стабілізатор напруги (LM2596S-3.3 P+)

Висока ефективність: Стабілізатор має високу ефективність до 90%, що знижує тепловиділення і підвищує надійність системи.

Простота використання: Легко інтегрується у схему завдяки стандартному корпусу та мінімальній кількості зовнішніх компонентів, що полегшує проектування та монтаж.

2.3 Робота із базами даних

Вибір сторонніх баз даних

Чому потрібні сторонні бази даних:

Сторонні бази даних, такі як Common Voice від Mozilla, надають великий набір даних для тренування моделей розпізнавання мовлення. Це дозволяє покращити якість та точність розпізнавання.

Обґрунтування вибору:

Common Voice - це відкрита база даних, яка містить аудіозаписи з транскрипціями на багатьох мовах, включаючи українську. Це допоможе тренувати модель для кращого розпізнавання української мови.

Google Speech API та інші комерційні сервіси також можуть бути використані для порівняння результатів та визначення найкращого підходу.

Формування власних баз даних

Процедура формування: Збір аудіозаписів розмов у різних умовах (фонового шуму, різних акцентів). Анотування аудіозаписів, тобто створення транскрипцій для кожного запису. Зберігання аудіозаписів та транскрипцій у структурованому форматі для подальшого використання в тренуванні моделей.

Характеристики баз даних:

Формат файлів: WAV або інший стандартний аудіо формат.

Метадані: мітки часу, умови запису, ідентифікатор мовця, транскрипції.

Обсяг: мінімум кілька сотень годин аудіозаписів для забезпечення якості моделі.

Проектування схеми бази даних:

UML модель системи(рисунок 4)

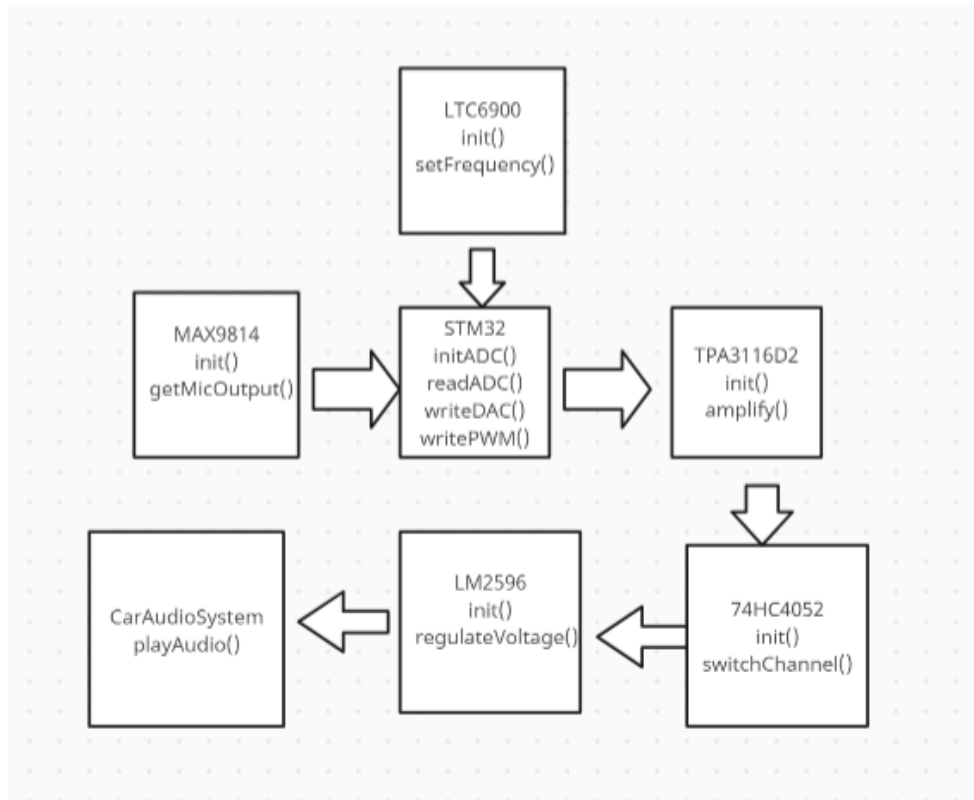


Рисунок 14 - UML схема послідовності (Sequence Diagram)

Взаємодія між класами:

MAX9814 передає оцифрований аудіосигнал до STM32F407 через метод readADC().

LTC6900 передає сигнал шуму до STM32F407 через метод writePWM().

STM32F407 передає аудіосигнал до TPA3116D2 через метод writeDAC().

74HC4052 перемикає аудіосигнал між штатним сигналом автомобіля та сигналом шуму за допомогою switchChannel().

TPA3116D2 підсилює аудіосигнал для динаміків автомобіля через метод amplify().

LM2596 забезпечує стабілізоване живлення для всіх компонентів системи.

CarAudioSystem отримує вихідний аудіосигнал для відтворення через динаміки автомобіля.

2.4 Висновок

У цьому розділі проведена розробка структурної схеми приладу. Описано блок-схему роботи приладу, обрано та обґрунтовано елементну базу, зображено зовнішній вигляд всіх елементів.

Схематично зображено підключення обраного контролера STM32F407VGTx до USB-to-Serial адаптеру на базі чіпу FT232RL. Проведена робота з базами даних та розроблено UML-схему послідовності(Sequence Diagram). Описано взаємодію між класами, та проведено проектування пристрою.

3 РЕАЛІЗАЦІЯ ПРИСТРОЮ ЗАХИСТУ МОВНОЇ ІНФОРМАЦІЇ ДЛЯ ПРОВЕДЕННЯ КОНФІДЕНЦІЙНИХ РОЗМОВ У АВТОМОБІЛІ

3.1 Обґрунтування вибору мови програмування

Для розробки коду було обрано мову програмування C. Мова C забезпечує прямий доступ до апаратних ресурсів мікроконтролера, що є критично важливим для систем реального часу.

Висока продуктивність: Завдяки ефективному використанню ресурсів та мінімальній накладній витраті, мова C є ідеальною для вбудованих систем.

Стандарт для вбудованих систем: Мова C є де-факто стандартом для розробки програмного забезпечення для мікроконтролерів та вбудованих систем, завдяки чому існує багато готових бібліотек і прикладів коду.

Портативність: Код на C можна легко перенести на інші мікроконтролери, що полегшує повторне використання коду в різних проектах.

Середовище розробки: STM32CubeIDE (рисунк 1)

Інтегроване середовище: STM32CubeIDE об'єднує редактор коду, компілятор, налагоджувач та інші інструменти, що полегшує розробку та налагодження програм для мікроконтролерів STM32.

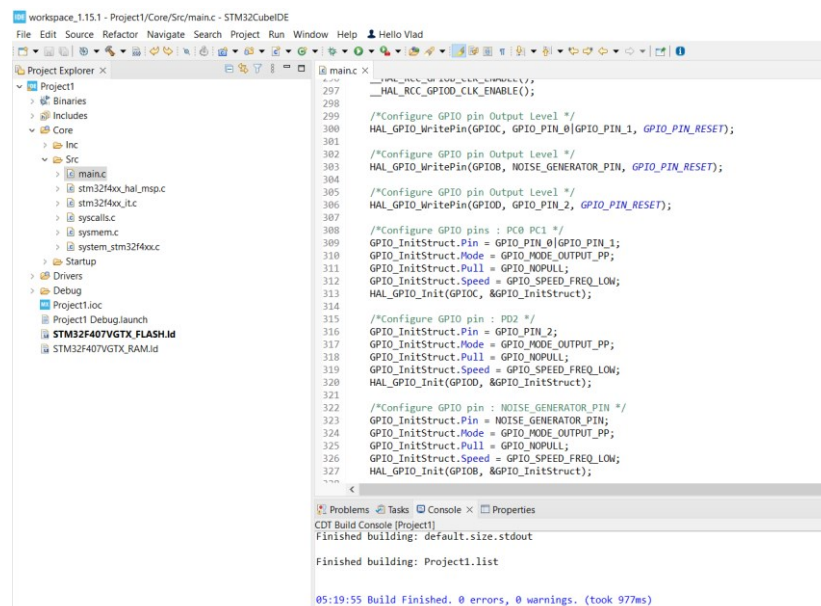


Рисунок 15 - Вигляд програмного середовища STM32CubeIDE

Графічний конфігуратор: STM32CubeMX(рисунок 1), інтегрований у STM32CubeIDE, дозволяє графічно налаштовувати периферійні пристрої мікроконтролера, що знижує ймовірність помилок при конфігурації.

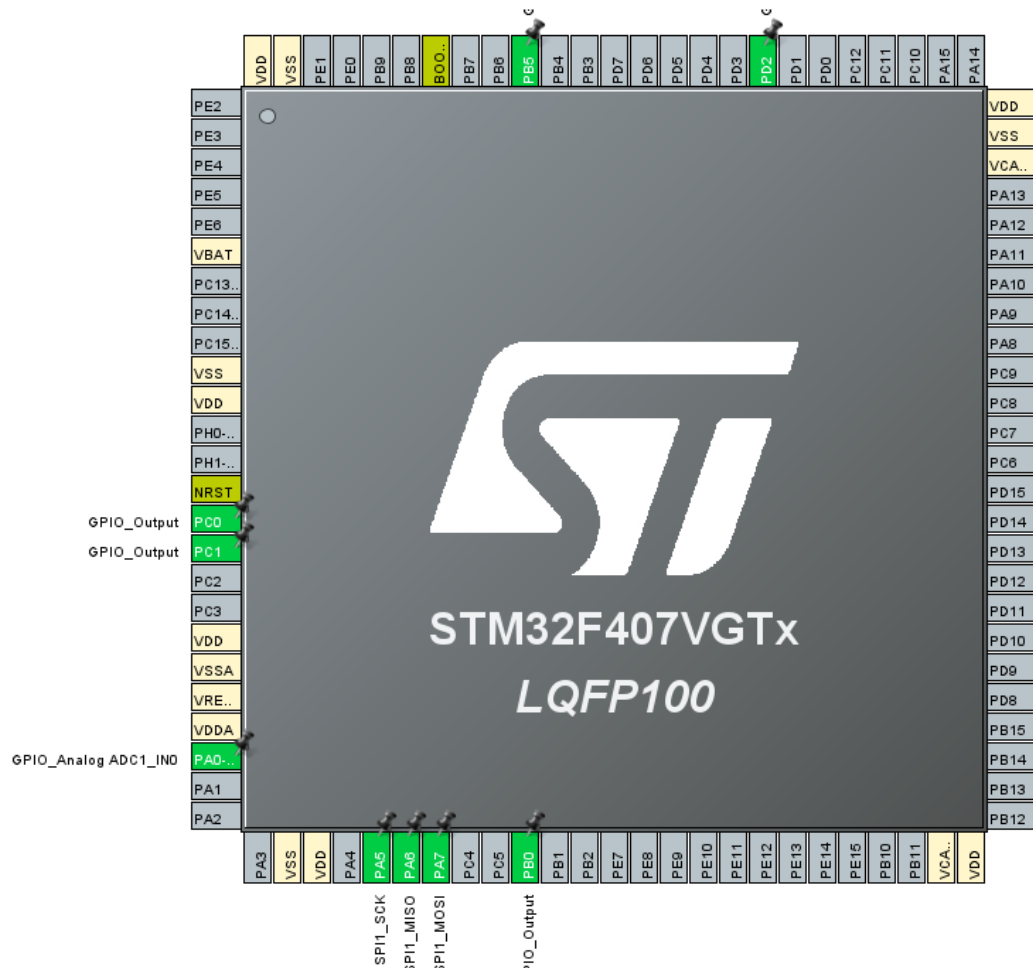


Рисунок 16 - Вигляд графічного конфігуратору у STM32CubeMX

Підтримка HAL бібліотек: STM32CubeIDE надає доступ до HAL (Hardware Abstraction Layer) бібліотек, що спрощує розробку, роблячи її більш абстрагованою від конкретного апаратного забезпечення.

Документація та підтримка: Вбудована документація, приклади коду та активна спільнота користувачів роблять STM32CubeIDE зручним середовищем для розробки.

3.2 Опис фрагментів лістингу програми

Конфігурація

```
#define NOISE_GENERATOR_PIN GPIO_PIN_5
#define NOISE_GENERATOR_PORT GPIOB
#define KEYWORD "confidential"
#define STOPWORD "stop"
```

Ці директиви визначають пін, до якого підключений генератор шуму, та слова, які шукаються у тексті, перетвореному з аудіо.

Змінні

```
ADC_HandleTypeDef hadc1;
SPI_HandleTypeDef hspi1;
UART_HandleTypeDef huart2;
uint8_t audio_buffer[256];
char audio_text[256];
char uart_buffer[UART_BUFFER_SIZE];
```

Ці змінні використовуються для зберігання станів периферійних пристроїв (ADC, SPI, UART) та для буферизації аудіо даних і тексту.

Функції

```
void Capture_Audio(void) {
    strcpy((char*)audio_buffer, "confidential conversation happening now");
    strcpy(audio_text, (char*)audio_buffer);
}
```

Функція `Capture_Audio` імітує захоплення аудіо даних та їх перетворення в текст.

```
int Analyze_Audio_For_Keyword(void) {
    return (strstr(audio_text, KEYWORD) != NULL);
}
```

Функція `Analyze_Audio_For_Keyword` перевіряє наявність ключового слова в тексті.

```
int Analyze_Audio_For_Stopword(void) {
    return (strstr(audio_text, STOPWORD) != NULL);
}
```

Функція `Analyze_Audio_For_Stopword` перевіряє наявність стоп слова в тексті.

```
void Start_Noise_Generator(void) {
```

```

    HAL_GPIO_WritePin(NOISE_GENERATOR_PORT, NOISE_GENERATOR_PIN, GPIO_PIN_SET);
}

```

Функція `Start_Noise_Generator` включає генератор шуму.

```

void Stop_Noise_Generator(void) {
    HAL_GPIO_WritePin(NOISE_GENERATOR_PORT, NOISE_GENERATOR_PIN, GPIO_PIN_RESET);
}

```

Функція `Stop_Noise_Generator` вимикає генератор шуму.

```

void UART_TransmitString(char *str) {
    HAL_UART_Transmit(&huart2, (uint8_t *)str, strlen(str), HAL_MAX_DELAY);
}

```

Функція `UART_TransmitString` відправляє строку через UART.

Головна функція

```

int main(void) {
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_SPI1_Init();
    MX_USART2_UART_Init();

    while (1) {
        Capture_Audio();

        if (Analyze_Audio_For_Keyword()) {
            Start_Noise_Generator();
            UART_TransmitLine("Keyword detected. Starting noise generator for 1 minute.");
            HAL_Delay(60000); // Шум протягом 1 хвилини

            Stop_Noise_Generator();
            UART_TransmitLine("Noise generator stopped. Waiting for 5 seconds.");
            HAL_Delay(5000); // Перерва 5 секунд

            Capture_Audio();

            if (!Analyze_Audio_For_Stopword()) {
                Start_Noise_Generator();
                UART_TransmitLine("Stop word not detected. Continuing noise for 1 more minute.");
                HAL_Delay(60000); // Продовження шуму ще на 1 хвилину
                Stop_Noise_Generator();
            }
        }
    }
}

```

```

    }
  }
}
}

```

Ця функція ініціалізує периферійні пристрої та запускає нескінченний цикл, в якому захоплюється аудіо, перевіряється наявність ключового слова, включається та вимикається генератор шуму з відповідними затримками. Всі важливі події логуються через UART.

Після виконання build “debug” помилок у коді виявлено не було, що означає що можна прошивати контролер.

```

Finished building: Project1.list

05:03:43 Build Finished. 0 errors, 0 warnings. (took 3s.517ms)

```

Рисунок 17 - Результат виконання “debug” для мого коду

Але через відсутність реального ST-Link прошивка неможлива.

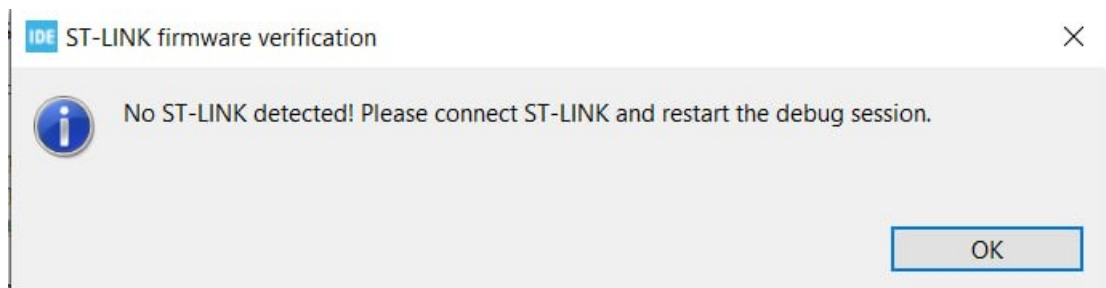


Рисунок 18 - Причина зупинення роботи з розробленим пристроєм

Для альтернативного вирішення проблеми я спростив код який використовувався і налаштував його для демонстраційного показу використовуючи вже готові пристрої, не змінюючи суті коду.

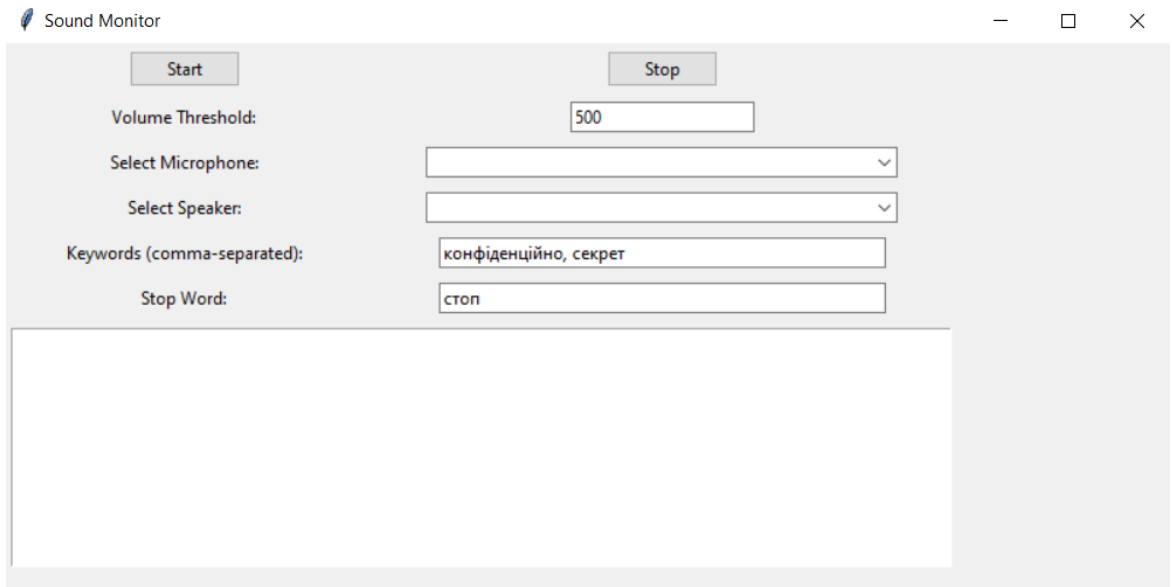


Рисунок 19 - Скріншот вигляду інтерфейсу користувача

Фрагменти лістингу програми

Імпорт бібліотек

```
import pyaudio
import wave
import audioop
import threading
import time
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext
import speech_recognition as sr
import pygame
```

Цей фрагмент коду імпортує необхідні бібліотеки:

pyaudio для обробки аудіо потоків.

wave для роботи з аудіофайлами.

audioop для обробки аудіоданих.

threading для створення багатопоточних додатків.

time для роботи з часом.

tkinter для створення графічного інтерфейсу.

speech_recognition для розпізнавання мови.

pygame для відтворення аудіофайлів.

Клас AudioMonitorApp

```

class AudioMonitorApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Audio Monitor")

        self.mic = None
        self.speaker = None
        self.threshold = 1000
        self.keywords = []
        self.stop_word = ""
        self.noise_file = "noise.wav"

        self.audio_interface = pyaudio.PyAudio()
        self.recognizer = sr.Recognizer()

        self.create_widgets()

```

Клас `AudioMonitorApp` відповідає за створення графічного інтерфейсу та основну логіку роботи програми:

Ініціалізує основні параметри, такі як вибір мікрофона, динаміка, поріг гучності, ключові слова та стоп-слово.

Створює екземпляри `pyaudio.PyAudio` для роботи з аудіо та `speech_recognition.Recognizer` для розпізнавання мови.

Викликає метод `create_widgets()` для створення віджетів інтерфейсу.

Метод `create_widgets`

```

def create_widgets(self):
    # Створення віджетів для інтерфейсу
    self.mic_label = tk.Label(self.root, text="Select Microphone:")
    self.mic_label.grid(row=0, column=0, padx=10, pady=10)

    self.mic_dropdown = ttk.Combobox(self.root, values=self.get_microphones())
    self.mic_dropdown.grid(row=0, column=1, padx=10, pady=10)
    self.mic_dropdown.current(0)

    self.speaker_label = tk.Label(self.root, text="Select Speaker:")
    self.speaker_label.grid(row=1, column=0, padx=10, pady=10)

    self.speaker_dropdown = ttk.Combobox(self.root, values=self.get_speakers())
    self.speaker_dropdown.grid(row=1, column=1, padx=10, pady=10)
    self.speaker_dropdown.current(0)

    self.threshold_label = tk.Label(self.root, text="Volume Threshold:")
    self.threshold_label.grid(row=2, column=0, padx=10, pady=10)

    self.threshold_slider = tk.Scale(self.root, from_=0, to=5000, orient=tk.HORIZONTAL)
    self.threshold_slider.set(self.threshold)
    self.threshold_slider.grid(row=2, column=1, padx=10, pady=10)

    self.keywords_label = tk.Label(self.root, text="Keywords:")

```

```

self.keywords_label.grid(row=3, column=0, padx=10, pady=10)

self.keywords_entry = tk.Entry(self.root)
self.keywords_entry.grid(row=3, column=1, padx=10, pady=10)

self.stop_word_label = tk.Label(self.root, text="Stop Word:")
self.stop_word_label.grid(row=4, column=0, padx=10, pady=10)

self.stop_word_entry = tk.Entry(self.root)
self.stop_word_entry.grid(row=4, column=1, padx=10, pady=10)

self.start_button = tk.Button(self.root, text="Start Monitoring", command=self.start_monitoring)
self.start_button.grid(row=5, column=0, padx=10, pady=10)

self.stop_button = tk.Button(self.root, text="Stop Monitoring", command=self.stop_monitoring)
self.stop_button.grid(row=5, column=1, padx=10, pady=10)

self.text_display = scrolledtext.ScrolledText(self.root, width=40, height=10)
self.text_display.grid(row=6, column=0, columnspan=2, padx=10, pady=10)

```

Цей метод створює віджети інтерфейсу:

Мітки та випадаючі списки для вибору мікрофона та динаміка.

Повзунок для налаштування порогу гучності.

Поля вводу для ключових слів та стоп-слова.

Кнопки для початку та зупинки моніторингу.

Поле для відображення тексту.

Метод start_monitoring

```

def start_monitoring(self):
    self.mic = self.mic_dropdown.get()
    self.speaker = self.speaker_dropdown.get()
    self.threshold = self.threshold_slider.get()
    self.keywords = self.keywords_entry.get().split(',')
    self.stop_word = self.stop_word_entry.get()

    self.is_running = True
    self.text_display.insert(tk.END, "Monitoring started...\n")

    self.monitor_thread = threading.Thread(target=self.monitor_audio)
    self.monitor_thread.start()

```

Метод start_monitoring запускає процес моніторингу:

Зберігає вибрані мікрофон, динамік, поріг гучності, ключові слова та стоп-слово.

Встановлює прапорець is_running у True.

Додає повідомлення в текстове поле.

Запускає потік для моніторингу аудіо.

Метод `monitor_audio`

```
def monitor_audio(self):
    stream = self.audio_interface.open(format=pyaudio.paInt16,
                                       channels=1,
                                       rate=44100,
                                       input=True,
                                       frames_per_buffer=1024,
                                       input_device_index=self.get_microphone_index(self.mic))

    while self.is_running:
        data = stream.read(1024)
        rms = audioop.rms(data, 2)

        if rms > self.threshold:
            self.process_audio(data)

    stream.stop_stream()
    stream.close()
```

Метод `monitor_audio` виконує моніторинг аудіо:

Відкриває аудіопотік з вибраного мікрофона.

В циклі зчитує аудіодані та обчислює середньоквадратичне значення (RMS) для визначення гучності.

Якщо гучність перевищує поріг, викликає метод `process_audio` для обробки аудіо.

Зупиняє та закриває аудіопотік після завершення моніторингу.

Метод `process_audio`

```
def process_audio(self, data):
    wav_file = wave.open("temp.wav", "wb")
    wav_file.setnchannels(1)
    wav_file.setsampwidth(self.audio_interface.get_sample_size(pyaudio.paInt16))
    wav_file.setframerate(44100)
    wav_file.writeframes(data)
    wav_file.close()

    with sr.AudioFile("temp.wav") as source:
        audio = self.recognizer.record(source)

    try:
        text = self.recognizer.recognize_google(audio).lower()
        self.text_display.insert(tk.END, f"Recognized: {text}\n")

        if any(keyword.lower() in text for keyword in self.keywords):
            self.play_noise()
    except sr.UnknownValueError:
        self.text_display.insert(tk.END, "Google Speech Recognition could not understand audio\n")
    except sr.RequestError as e:
```

```
self.text_display.insert(tk.END, f"Could not request results from Google Speech Recognition service; {e}\n")
```

Метод `process_audio` обробляє аудіодані:

Записує аудіодані у тимчасовий WAV файл.

Використовує бібліотеку `speech_recognition` для розпізнавання мови з тимчасового файлу.

Вставляє розпізнаний текст у текстове поле.

Перевіряє наявність ключових слів у розпізаному тексті.

Якщо ключові слова виявлені, викликає метод `play_noise` для відтворення білого шуму.

Метод `play_noise`

```
def play_noise(self):
    pygame.mixer.init()
    pygame.mixer.music.load(self.noise_file)
    pygame.mixer.music.play(-1)

    end_time = time.time() + 60 # 60 секунд шуму
    while time.time() < end_time:
        if self.check_stop_word():
            break
        time.sleep(0.1)

    pygame.mixer.music.stop()
    time.sleep(10) # пауза 10 секунд

def check_stop_word(self):
    with sr.Microphone() as source:
        audio = self.recognizer.listen(source, timeout=10)
    try:
        text = self.recognizer.recognize_google(audio).lower()
        self.text_display.insert(tk.END, f"Recognized during noise: {text}\n")
        if self.stop_word.lower() in text:
            self.text_display.insert(tk.END, "Stop word detected, stopping noise.\n")
            return True
    except sr.UnknownValueError:
        self.text_display.insert(tk.END, "Could not understand audio during noise.\n")
    except sr.RequestError as e:
        self.text_display.insert(tk.END, f"Could not request results during noise; {e}\n")
    return False
```

Методи `play_noise` та `check_stop_word` відтворюють білий шум і перевіряють наявність стоп-слова:

Ініціалізують `pygame.mixer` для відтворення білого шуму з файлу.

Відтворюють білий шум у циклі протягом 60 секунд.

Використовують метод `check_stop_word` для перевірки наявності стоп-слова під час відтворення шуму.

Якщо стоп-слово виявлене, зупиняють відтворення шуму.

Після відтворення шуму викликають паузу на 10 секунд перед наступним циклом моніторингу.

3.3 Тестування програмного засобу для автоматичного захисту мовної інформації

Для демонстрації тестування програмного засобу(рисунок 8) і надання кількісних показників, потрібно провести серію тестів, які перевіряють функціональність, продуктивність та надійність системи.

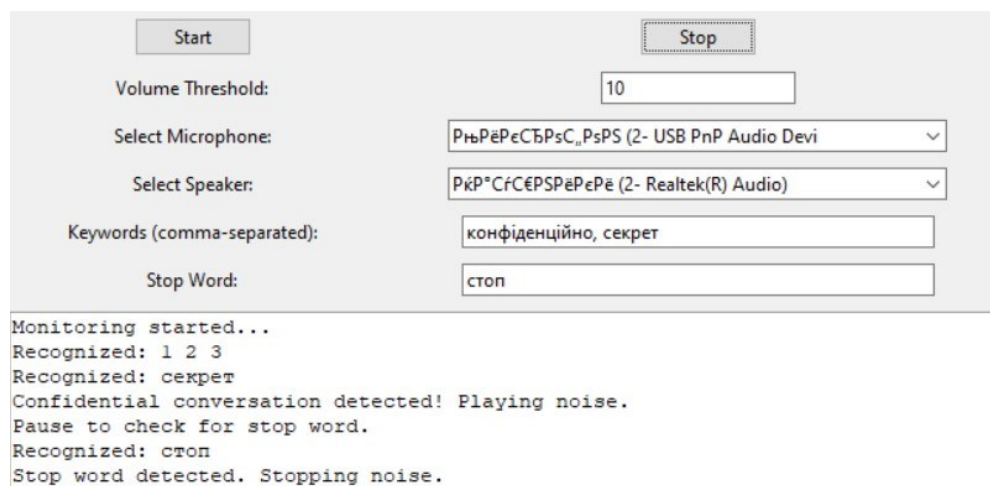


Рисунок 20 - Робота програмного засобу

Функціональне тестування перевіряє, чи працює система відповідно до специфікацій.

Тести:

Виявлення звуку: Перевірка, чи система правильно виявляє звуки вище порогового значення.

Аналіз звуку: Перевірка, чи система правильно розпізнає ключові слова.

Активація захисту: Перевірка, чи система активує захист при виявленні конфіденційної розмови.

Кількісні показники:

Точність виявлення звуку: відсоток правильного виявлення звуків вище порогового значення.

Точність розпізнавання ключових слів: відсоток правильного розпізнавання ключових слів.

Час активації захисту: середній час від виявлення ключового слова до активації захисту.

Продуктивність тестування

Продуктивність тестування перевіряє, як система виконує завдання під навантаженням.

Тести:

Тест навантаження: Виявлення здатності системи обробляти велику кількість запитів.

Тест на пропускну здатність: Вимірювання кількості звуків, які система може обробляти за одиницю часу.

Кількісні показники:

Пропускна здатність: кількість звуків, оброблених за секунду.

Середній час відповіді: середній час відповіді системи на події.

3. Надійність тестування

Надійність тестування перевіряє стабільність системи протягом тривалого часу.

Тести:

Тест на тривалу стабільність: Перевірка стабільності системи при тривалій роботі.

Тест на відмовостійкість: Перевірка здатності системи відновлюватись після збоїв.

Кількісні показники:

Час безвідмовної роботи: середній час роботи системи без збоїв.

Час відновлення: середній час, потрібний для відновлення після збою.

Результати тестування:

1. Точність виявлення звуку:

Розрахунок:

Точність виявлення звуку = (Кількість правильно виявлених звуків / Загальна кількість звуків) * 100%

Для тесту було взято речення: для демонстрації тестування програмного засобу і надання кількісних показників, потрібно провести серію тестів, які перевіряють функціональність, продуктивність та надійність системи.

Результат:

Recognized: для демонстрації тестування програмного засобу і надання кількісних показників потрібно провести серію тестів які перевіряють функціональність продуктивність та

Рисунок 21 - Результат тесту на точність виявлення звуку

Загальна кількість звуків - 156

Кількість правильно виявлених звуків - 140

Точність виявлення звуку = 89,74%

2. Точність розпізнавання ключових слів

Розрахунок:

Точність розпізнавання ключових слів = (Кількість правильно розпізнаних ключових слів / Загальна кількість ключових слів) * 100%

Для тесту було взято слова: секрет, секрети, секретно, секретна, засекречено, засекречена, таємно, таємна, таємниця, таємниці, цілком таємно, конфіденційна, конфіденційно.

Результат:

Recognized: секрет секрети секретно секретна
Recognized: засекречено засекречено таємна
Recognized: таємниця таємниці цілком таємно конфіденційна
Recognized: конфіденційна

Рисунок 22 - Результат тесту на точність розпізнавання ключових слів

Загальна кількість ключових слів - 13

Кількість правильно розпізнаних ключових слів - 10

Точність розпізнавання ключових слів = 76,9%

3. Час активації захисту

Час від моменту виявлення ключового слова до моменту активації захисту (відтворення білого шуму): 0,1с

4. Пропускна здатність

Розрахунок:

Пропускна здатність = (Загальна кількість оброблених звуків / Час у секундах)

Загальна кількість оброблених звуків - 362

Час у секундах - 39,17

Пропускна здатність = 9,241

5. Середній час відповіді

Середній час відповіді = (Сума часу відповіді на всі події / Кількість подій):10 секунд

6. Час безвідмовної роботи

Розрахунок:

Час безвідмовної роботи: під час роботи збоїв не виникло

7. Час відновлення

Проведено 5 випадків штучного збою роботи та зафіксовано час на відновлення.

Час необхідний для відновлення: 13,89 12,08 13,10 12,69 14,02 секунд.

Час відновлення = (Сума часу відновлення після всіх збоїв / Кількість збоїв)

Сума часу відновлення після всіх збоїв - 65,78 секунд

Кількість збоїв - 5

Час відновлення = 13,156 секунд

Таблиця 3. Результати тестування

Тест	Результат
Точність виявлення звуку	89,74%
Точність розпізнавання ключових слів	76,9%

Час активації захисту	0,1 секунд
Пропускна здатність	9,241 знаків/секунду
Середній час відповіді	10 секунд
Час безвідмовної роботи	100%
Час відновлення	13,156 секунд

Також варто провести тест функціональності генератору шуму. Для цього після активації захисту проведено заміри гучності за допомогою програми SoundMeter(рисунок 11).

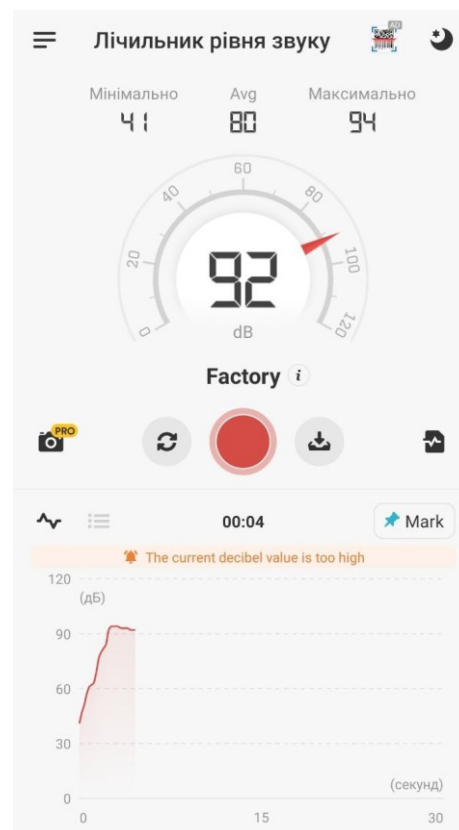


Рисунок 23 - Вимірювання гучності шуму зовнішніх динаміків

Рівень гучності білого шуму повинен перевищувати рівень звукового тиску розмови, яку потрібно приховати.

Також проведено тест гучності вбудованого динаміку ноутбуку.

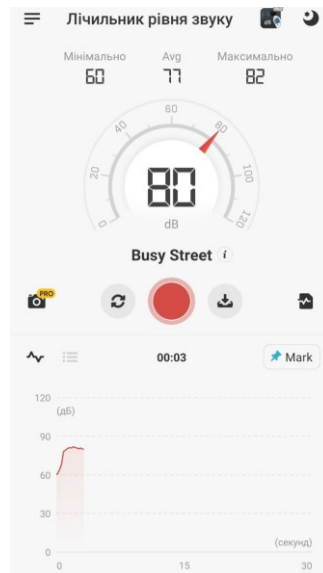


Рисунок 24 - Вимірювання гучності шуму вбудованого динаміку

Зазвичай розмова має рівень близько 60-65 дБ, тому білий шум на рівні 70-90 дБ буде ефективним для маскування без створення надмірного шумового навантаження.

Отже з отриманих результатів можна зробити висновок що данна система захисту мовної інформації може бути успішно використана у реальних умовах під час конфіденційної розмови у автомобілі з метою захисту інформації від посторонніх осіб.

3.4. Висновок

Для розробки коду було обрано мову програмування C. Мова C забезпечує прямий доступ до апаратних ресурсів мікроконтролера, що є критично важливим для систем реального часу.

Інтегроване середовище: STM32CubeIDE об'єднує редактор коду, компілятор, налагоджувач та інші інструменти, що полегшує розробку та налагодження програм для мікроконтролерів STM32.

Наведено фрагменти лістингу програми.

Результат тестування показали, що данна система захисту мовної інформації може бути успішно використана у реальних умовах під час конфіденційної розмови у автомобілі з метою захисту інформації від посторонніх осіб.

ВИСНОВОК

В даній бакалаврській дипломній роботі схематично спроектовано та розроблено прилад який реалізовує систему захисту мовної інформації для проведення конфіденційних розмов в автомобілі та протестовано його програмну частину, здійснено програмну реалізацію засобу для захисту конфіденційної інформації.

У першому розділі проведено всебічний аналіз стану питання розробки систем захисту мовної інформації для проведення конфіденційних розмов в автомобілі, порівняльний аналіз існуючих аналогів та аналіз методів розв'язання цієї задачі.

Загалом, сучасні технології пропонують широкий спектр рішень для захисту мовної інформації в автомобілях.

Другий розділ був зосереджений на розробці та проектуванні пристрою який реалізовує систему захисту мовної інформації для проведення конфіденційної розмови.

Обґрунтовано вибір мови програмування та елементів пристрою, які використано.

Проведено детальний опис фрагментів лістингу програми, які реалізує алгоритм.

Вдало протестовано програмний засіб, на скріншотах та таблиці наведено результати випробовувань та як працює засіб.

З отриманих результатів можна зробити висновки що розроблений пристрій може бути використано при конфіденційних розмовах для забезпечення захисту інформації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Кушнерьов О. С. БЕЗПЕКА ІНФОРМАЦІЇ: Конспект лекцій - Сумський державний університет СДУ, 2021. - 15 с
2. Кушнерьов О. С. БЕЗПЕКА ІНФОРМАЦІЇ: Конспект лекцій - Сумський державний університет СДУ, 2021. - 31 с
3. Павлов О.А. Аналіз безпеки автомобіля на основі моделі загроз: Магістерська дисертація - Київ, 2018. - 22 с
4. Павлов О.А. Аналіз безпеки автомобіля на основі моделі загроз: Магістерська дисертація - Київ, 2018. - 37 с
5. Hemi Anechoic Chambers [Електронний ресурс]: <https://www.iacacoustics.global/news/the-difference-between-hemi-anechoic-chambers-and-fully-anechoic-chambers/>
6. Кашканов А.А., Кужель В.П., Грисюк О.Г. Інформаційні комп'ютерні системи автомобільного транспорту: Навчальний посібник - Вінниця ВНТУ, 2010. - 9 с
7. Конахович Г.Ф., Климчук В.П., Паук С.М., Потапов В.Г., Чуприн В.М., Горбунов О.О. Захист інформації в телекомунікаційних системах - Навчальний посібник - Київ, 2009. - 58 с
8. Чеканін О.Ю., Жданова О.Г. Модель загроз для оцінки безпеки автомобіля - Матеріали науково-практичної конференції «Інформатика та обчислювальна техніка ІОТ-2018». – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського».
9. K. Koscher et al., “Experimental security analysis of a modern automobile,” in Proceedings — IEEE Symposium on Security and Privacy, 2010, pp. 447–462
10. E. Nickel, “IBM automotive software foundry,” in Press Conference on Computer Science in Automotive Industry, Frankfurt University, Frankfurt, Germany, September 2003.

11. Азарова О.А., к.т.н., Біліченко Н.О Гібридний засіб захисту мовної інформації - Стаття Вінницький національний технічний університет, ВНТУ, 2020.

12. Катаєв В. С., Сінюгін В. В., Грицак А. В., Павловський П. В. Мобільний засіб блокування витоку інформації акустичними каналами - науково-технічний журнал, Вінницький національний технічний університет, ВНТУ- 2021.

13. Технічні канали витоку інформації. Порядок створення комплексів технічного захисту інформації [Електронний ресурс] : навчальний посібник / С. О. Іванченко, О. В. Гавриленко, О. А. Липський [та ін.] ; - Київ : НТУУ «КПІ», 2016. - 104 с. Режим доступу: https://ela.kpi.ua/handle/1234_56789/15155.

14. Яремчук Ю.Е. Дослідження характеристик різних типів шумів для захисту інформації від витоку лазерним каналом. / Катаєв В.С., Яремчук Ю.Е., Сінюгін В.В. // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. - Випуск 2(32), 2016. - С. 21-27.

15. Яремчук Ю.Е. Метод активного захисту інформації від зняття лазерними системами акустичної

розвідки / Катаєв В.С., Яремчук Ю.Е. // Захист інформації. - Т. 21, №1, 2019. - С. 34-39.

16. Остапов С.Е. Технології захисту інформації : навчальний посібник / С.Е. Остапов, С. П. Евсеев, О.Г. Король. -Х.: Вид. ХНЕУ, 2013. -476с.

17. Пристрій забезпечення конфіденційних переговорів DRUID D-06 [Електронний ресурс]. – Режим доступу : <https://one-click.com.ua/uk/ustroystvo-obespecheniya-konfidencialnyh-peregovorov-druid-d-06.html>

18. Проблеми активного захисту інформації від витоку через віброакустичні канали / В.С. Катаєв, А.В. Грицак, В.О. Леонтєв, Н.В. Ляховченко // Реєстрація, зберігання і обробка даних. - Т. 18, N23, 2016. - С. 54.

19. Яремчук Ю.Е. Можливості практичного застосування тепловізорів у питаннях захисту інформації /

Ю.Е. Яремчук, В.С. Катаєв, М.Ю. Гижко, С.М. Скуратов // Правове, нормативне та метрологічне забезпечення

системи захисту інформації в Україні. - Випуск 1(31), 2016. - С. 99-105.

20. Мобільний генератор шуму MNG-300 Rabbler [Електронний ресурс]. – Режим доступу: <https://www.forter.com.ua/poiskovaya-tehnika/mng-300-rabbler/>

21. COCOON Secure [Електронний ресурс]. – Режим доступу: <https://www.cyber-cocoon.eu/objectives/>

22. Хилобокий С.М. «Розробка мобільного засобу блокування інформації від витоку прямим акустичним каналом для проведення конфіденційних розмов у незахищеному середовищі»: Дипл. робота., фак-т менедж. та інф. безпеки, ВНТУ, Вінниця, 2021.

23. Clear Voice [Електронний ресурс]. – Режим доступу: <https://helpguide.sony.net/icd/u54/v1/en1/contents/TP0000153601.html>

24. Michael R. Neuert, MA, BSME Instructions for the “Acousticom” [Електронний ресурс]. – Режим доступу: <https://emfcenter.com/wp-content/uploads/2020/05/Acousticom-2-RF-Meter-Instructions-A-12-14-15.pdf> с.10.

25. Fengjun Li, Kaitai Liang, Zhiqiang Lin, Sokratis K. Katsikas Security and Privacy in Communication Networks [Електронний ресурс]. – Режим доступу: <https://link.springer.com/book/10.1007/978-3-031-25538-0>

26. Speech Privacy Systems - Sound Masking Devices [Електронний ресурс]. – Режим доступу: <https://www.newportnetworksolutions.com/blog/how-sound-masking-works/>

27. Portable Cell Phone Jammers [Електронний ресурс]. – Режим доступу: <https://www.thesignaljammer.com/product-category/portable-cell-phone-jammers/>

28. Blackphone [Електронний ресурс]. – Режим доступу: <https://www.r-tt.com/technology-articles/what-is-blackphone.html>

29. Signal [Електронний ресурс]. – Режим доступу: <https://itvmg.com/news/u-chomu-osoblivist-signal-i-chomu-tsey-dodatok-bezpechnishiy-nizh-whatsapp-i-telegram-94084>

30. Mobile Soundproof Booths [Електронний ресурс]. – Режим доступу: <https://www.vocalboothtogo.eu/spb33s-mobile-soundproof-er--single-wall-sound-isolation-booth--solo-2/>

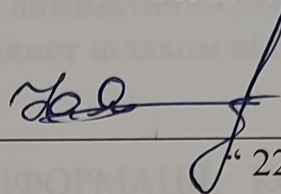
ДОДАТКИ

Додаток А - Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ

Голова секції “Управління інформаційною
безпекою” кафедри МБІС
д.т.н., професор



Юрій ЯРЕМЧУК

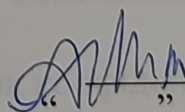
“ 22 ” березня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до бакалаврської дипломної роботи на тему:
Розробка системи захисту мовної інформації для проведення конфіденційних
розмов в автомобілі

08-72.БДР.018.00.048.ТЗ

Керівник бакалаврської дипломної роботи
к.ф.-м.н., доцент



Шиян А.А.

1. Найменування та область застосування

Розробка системи захисту мовної інформації для проведення конфіденційних розмов в автомобілі. Область застосування: захист мовної інформації для проведення конфіденційних розмов у автомобілі.

2. Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 80 від 11.03.2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка ефективної системи захисту мовної інформації для конфіденційної розмови.

3.2 Призначення: розроблений пристрій автоматично дізнається про початок конфіденційної розмови, і виконує захист шляхом відтворення шумових завіси.

4. Джерела розробки

4.1. Кушнерьов О. С. БЕЗПЕКА ІНФОРМАЦІЇ: Конспект лекцій - Сумський державний університет СДУ, 2021. - 15 с

4.2. Павлов О.А. Аналіз безпеки автомобіля на основі моделі загроз: Магістерська дисертація - Київ, 2018. - 22 с

4.3. Кашканов А.А., Кужель В.П., Грисюк О.Г. Інформаційні комп'ютерні системи автомобільного транспорту: Навчальний посібник - Вінниця ВНТУ, 2010. - 9 с

4.4. Конахович Г.Ф., Климчук В.П., Паук С.М., Потапов В.Г., Чуприн В.М., Горбунов О.О. Захист інформації в телекомунікаційних системах - Навчальний посібник - Київ, 2009. - 58 с

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Пристрій повинен відстежувати початок конфіденційної розмови і виконувати захист мовної інформації.

5.2 Вимоги до надійності:

5.2.1 Пристрій повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Пристрій повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам'ять – не менше 512 Mb;
 – середовище функціонування – операційна система сімейство Windows;
 – вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп'ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов'язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.3

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист мовної інформації під час конфіденційної розмови від несанкціонованого використання.

7.2 Неможливість отримання інформації про конфіденційну розмову через систему захисту для несанкціонованого використання.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

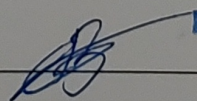
№ з/п	Назва етапів бакалаврської дипломної роботи	Початок	Закінчення
1.	Визначення напрямку бакалаврської роботи, формулювання теми		
2.	Аналіз предметної області обраної теми		
3.	Розробка алгоритму роботи		
4.	Написання бакалаврської роботи на основі розробленої теми		
5.	Передзахист бакалаврської дипломної роботи		
6.	Виправлення, уточнення, корегування бакалаврської дипломної роботи		
7.	Захист бакалаврської дипломної роботи		

10. Порядок контролю та прийому

10.1 До приймання бакалаврської дипломної роботи надається:

- ПЗ до бакалаврської дипломної роботи;
- демонстрація результату бакалаврської дипломної роботи;
- презентація;
- відзив керівника роботи;
- відзив рецензента

Технічне завдання до виконання прийняв



Рижук В.В.

Додаток Б – Лістинг програми

```

/* Includes -----*/
#include "main.h"
#include <string.h> // Додано для роботи зі строками

/* Private includes -----*/
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/
#define NOISE_GENERATOR_PIN GPIO_PIN_5
#define NOISE_GENERATOR_PORT GPIOB
#define KEYWORD "confidential"
#define STOPWORD "stop"

/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
ADC_HandleTypeDef hadc1;
SPI_HandleTypeDef hspi1;

/* USER CODE BEGIN PV */
uint8_t audio_buffer[256]; // Буфер для аудіо даних
char audio_text[256]; // Буфер для перетворення аудіо в текст (імітація)
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);
static void MX_SPI1_Init(void);
void Capture_Audio(void);
int Analyze_Audio_For_Keyword(void);
int Analyze_Audio_For_Stopword(void);
void Start_Noise_Generator(void);
void Stop_Noise_Generator(void);

/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */

void Capture_Audio(void)
{

```

```

// Імітація захоплення аудіо
// У реальній реалізації тут буде захоплення даних з ADC
strcpy((char*)audio_buffer, "confidential conversation happening now");
strcpy(audio_text, (char*)audio_buffer); // Імітація перетворення аудіо в текст
}

int Analyze_Audio_For_Keyword(void)
{
    // Імітація аналізу аудіо на наявність ключового слова
    return (strstr(audio_text, KEYWORD) != NULL);
}

int Analyze_Audio_For_Stopword(void)
{
    // Імітація аналізу аудіо на наявність стоп слова
    return (strstr(audio_text, STOPWORD) != NULL);
}

void Start_Noise_Generator(void)
{
    HAL_GPIO_WritePin(NOISE_GENERATOR_PORT, NOISE_GENERATOR_PIN, GPIO_PIN_SET);
}

void Stop_Noise_Generator(void)
{
    HAL_GPIO_WritePin(NOISE_GENERATOR_PORT, NOISE_GENERATOR_PIN, GPIO_PIN_RESET);
}

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_SPI1_Init();

```

```

/* USER CODE BEGIN 2 */

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    Capture_Audio();

    if (Analyze_Audio_For_Keyword())
    {
        Start_Noise_Generator();
        HAL_Delay(60000); // Шум протягом 1 хвилини

        Stop_Noise_Generator();
        HAL_Delay(5000); // Перерва 5 секунд

        Capture_Audio();

        if (!Analyze_Audio_For_Stopword())
        {
            Start_Noise_Generator();
            HAL_Delay(60000); // Продовження шуму ще на 1 хвилину
            Stop_Noise_Generator();
        }
    }
}
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Configure the main internal regulator output voltage
    */
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /** Initializes the RCC Oscillators according to the specified parameters
    * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
}

```

```

/** Initializes the CPU, AHB and APB buses clocks
 */
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief ADC1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_ADC1_Init(void)
{

    /* USER CODE BEGIN ADC1_Init 0 */

    /* USER CODE END ADC1_Init 0 */

    ADC_ChannelConfTypeDef sConfig = {0};

    /* USER CODE BEGIN ADC1_Init 1 */

    /* USER CODE END ADC1_Init 1 */

    /** Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)
    */
    hadc1.Instance = ADC1;
    hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV2;
    hadc1.Init.Resolution = ADC_RESOLUTION_12B;
    hadc1.Init.ScanConvMode = DISABLE;
    hadc1.Init.ContinuousConvMode = DISABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 1;
    hadc1.Init.DMAContinuousRequests = DISABLE;
    hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
    if (HAL_ADC_Init(&hadc1) != HAL_OK)
    {
        Error_Handler();
    }

    /** Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample
time.
    */
    sConfig.Channel = ADC_CHANNEL_0;
    sConfig.Rank = 1;
    sConfig.SamplingTime = ADC_SAMPLETIME_3CYCLES;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
    {

```

```

        Error_Handler();
    }
    /* USER CODE BEGIN ADC1_Init 2 */

    /* USER CODE END ADC1_Init 2 */

}

/**
 * @brief SPI1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_SPI1_Init(void)
{

    /* USER CODE BEGIN SPI1_Init 0 */

    /* USER CODE END SPI1_Init 0 */

    /* USER CODE BEGIN SPI1_Init 1 */

    /* USER CODE END SPI1_Init 1 */
    /* SPI1 parameter configuration*/
    hspi1.Instance = SPI1;
    hspi1.Init.Mode = SPI_MODE_MASTER;
    hspi1.Init.Direction = SPI_DIRECTION_2LINES;
    hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
    hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
    hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
    hspi1.Init.NSS = SPI_NSS_SOFT;
    hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_16;
    hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
    hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
    hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
    hspi1.Init.CRCPolynomial = 10;
    if (HAL_SPI_Init(&hspi1) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN SPI1_Init 2 */

    /* USER CODE END SPI1_Init 2 */

}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();

```

```

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0|GPIO_PIN_1, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOB, NOISE_GENERATOR_PIN, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_2, GPIO_PIN_RESET);

/*Configure GPIO pins : PC0 PC1 */
GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_1;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

/*Configure GPIO pin : PD2 */
GPIO_InitStruct.Pin = GPIO_PIN_2;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

/*Configure GPIO pin : NOISE_GENERATOR_PIN */
GPIO_InitStruct.Pin = NOISE_GENERATOR_PIN;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */

```

```
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

Додаток В - Презентація

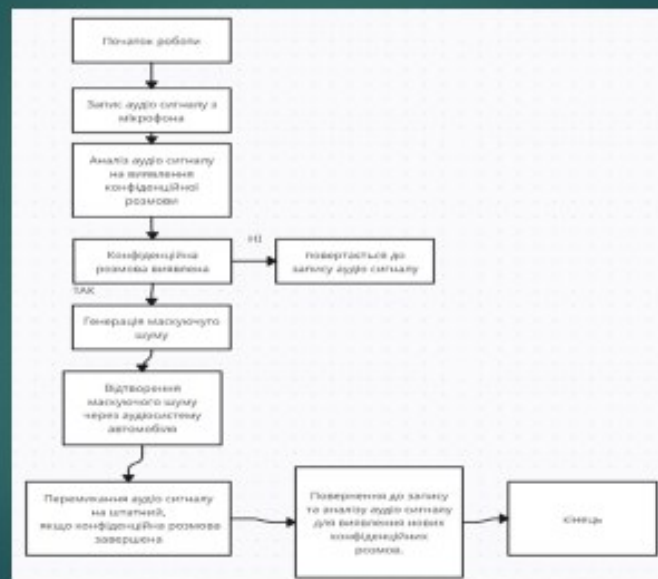
Вінницький національний технічний університет

Розробка системи захисту мовної
інформації дл проведення конфіденційної
розмови у автомобілі

Виконав:
студент 4 курсу
групи 2КІТС-206
Рижук В.В.

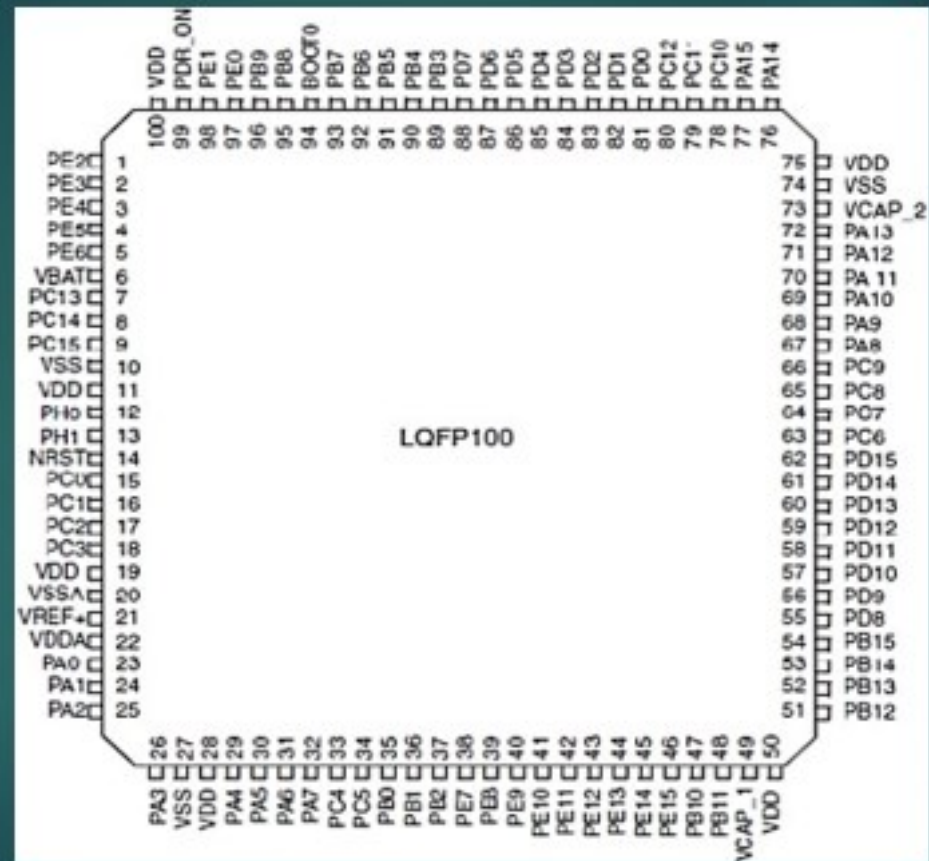
Вінниця 2024

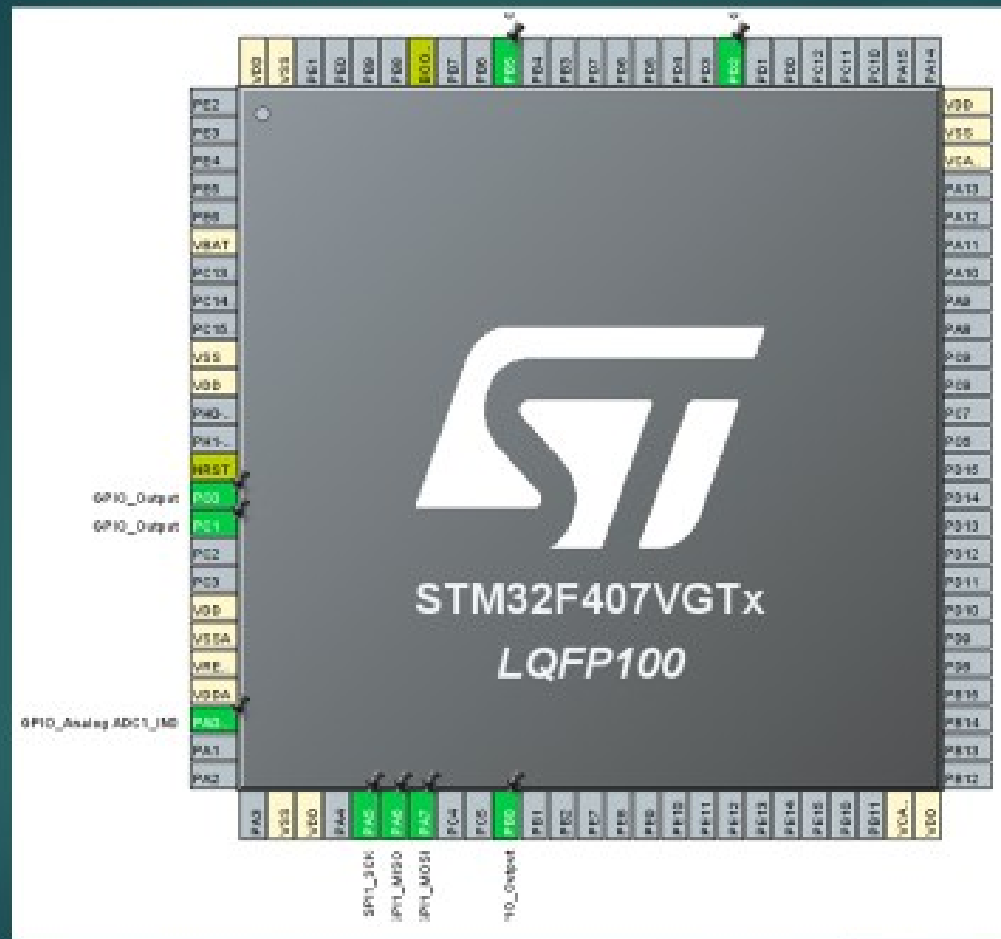
Блок схема роботи приладу

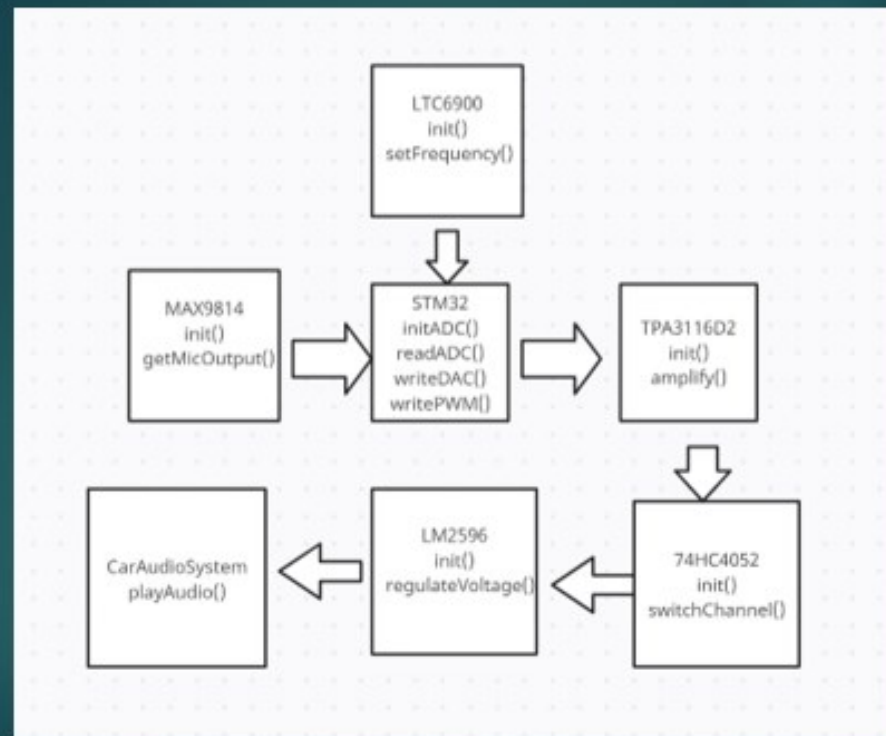


Мікроконтролер STM32F407VGTx









Start Stop

Volume Threshold: 10

Select Microphone: PnP*PnP*PnP*PnP*PnP (2- USB PnP Audio Devi

Select Speaker: PnP*PnP*PnP*PnP*PnP (2- Realtek(R) Audio)

Keywords (comma-separated): конфіденційно, секрет

Stop Word: стоп

Monitoring started...

Recognized: 1 2 3

Recognized: секрет

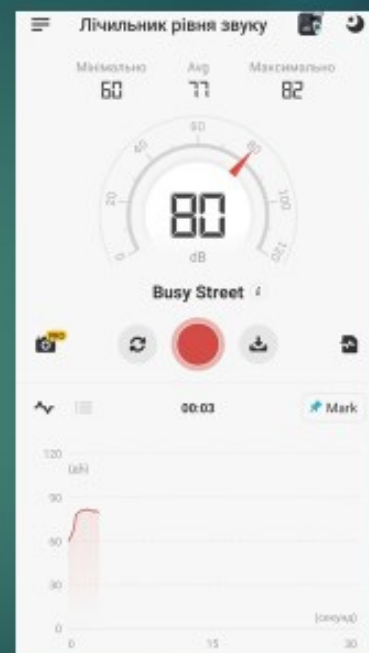
Confidential conversation detected! Playing noise.

Pause to check for stop word.

Recognized: стоп

Stop word detected. Stopping noise.

Тест	Результат
Точність виявлення звуку	89,74%
Точність розпізнавання ключових слів	76,9%
Час активації захисту	0,1 секунд
Пропускна здатність	9,241 знаків/секунду
Середній час відповіді	10 секунд
Час безвідмовної роботи	100%
Час відновлення	13,156 секунд





Дякую за увагу

Додаток В - Перевірка на автималярізм

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка системи захисту особистої інформації для проведення конфіденційних розмов в автомобілі

Тип роботи: бакалаврська дипломна робота

(напр. мкр)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем

Факультет менеджменту та інформаційної безпеки

(кафедра, факультет)

Показники звіту подібності Unichек

Оригінальність 96 %

Схожість 4 %

Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться наміски створення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____

(підпис)

Коваль Н.П.

(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichек щодо роботи.

Автор роботи

(підпис)

Рижук В.В.

(прізвище, ініціали)

Керівник роботи

(підпис)

Шиян А.А.

(прізвище, ініціали)